Passau University

Master Thesis

AUTOMATIC TEST CASE GENERATION TO EXPLORE PARAMETERS OF CRASH SCENARIOS IN SELF DRIVING CARS

Author:
Name Sameed Tariq

Supervisor:
Prof. Dr.-Ing. Gordon Fraser

Advisor:
Alessio Gambi, Ph.D.

Alessio up to here.

Monday 15th July, 2019

Abstract

The automotive industry is investing billion of dollars in self-driving cars that enable them to perceive the environment and drive the million of miles with range of driving scenarios. The promise of autonomous vehicles is to ensure the safety of roads, pedestrian and reduce the traffic congestion on roads. However, no rules are established yet for the certifying of self-driving car but the Governments are playing supportive role for the encouragement of autonomous car technologies. The intelligent system (car controlling software) must be able to handle the critical scenarios and foresee the environmental cars to avoid any misadventure before the position the cars in the public roads. Virtualization of the test cases in the simulated city (crash scenario) have been introduced to test the car in ~~infinite~~ number of scenarios to unveil the safety bugs. The search based testing techniques to explore parameters for finite cases shows promising result. The intention of the research is generate test cases for simulation of the car crash from Police Report. The direction of the work is to explore meaningful parameter in input scenarios and observed the behaviour of the car to restrict the test cases of crash scenario to interesting and relevant settings.

1 Introduction and Motivation

Self-Driving Cars are becoming one of the most profound technologies and the Auto Makers compete for the dominance of the technology[1]. Autonomous cars are becoming more intelligent and expected to replace the human drivers and increase the safety of roads by avoiding traffic accidents. The sales of autonomous cars is expected to reach 90$ Billion in 2030[2][3].There are several ongoing projects within this domain (e.g. Waymo [4]), most of them are advance enough to deploy on roads and under testing in everyday life. However, the development of fatal crashes in autonomous cars raises questions about the testing of controlling software and decision making in critical scenarios [5]. The testing of autonomous cars on public road is necessary for the development and enhancement of self driving technology because autonomous cars does not know how to behave with the unfamiliar situation or scenarios. Testing of the autonomous cars on real road for number of days to find the flaws is time consuming and tremendously challenging task.

The traditional way of testing the self-driving cars for naturalistic driving scenarios is Field Operational Tests (FOT). The downside of FOT that it won't be able to cover all the interesting scenarios and need to cover million of miles to test erroneous behavior and damage (failure) because of collision with car, pedestrian etc [6]. The relevant instance of autonomous car crashes are Waymo, Uber and Tesla Car crash. The Uber self-driving car [7] [8] failed to recognize an object such as women which caused it to ignore a pedestrian walking and classified as "false positives," that wouldn't be an issue for the vehicle, like a plastic bag or piece of paper. On the other side, Google self-driving car [9] hit the bus while changing the lane on the series of assumption of Google's program. The car anticipated that the bus will allow slow down and allow the car to cut in. Hence, it shows that the self-driving car are far away from handling the critical scenarios and sustainable testing system is required for the testing of autonomous cars to prevent the accidents and enhance the safety of roads and society.

The synthetic data is used for the testing of the autonomous cars in the virtual simulation. The car controlling software [10] is not tested with real data or realistic environment but with synthetic data or simulated data to test the software with different environmental conditions [11][12]. The distinct input in the self-driving cars with realistic scenarios in virtual testing will reduce the cost of physical testing. Indeed, the virtual testing can't replace the real-world testing but it narrow the gap in the testing of autonomous cars [13]. The creation of realistic scenarios for the simulator is the major issue and the creation of realistic maps which has to be done manually is a time consuming process [14].

The NHTSA (National Highway Traffic Safety Administration) [15] formulated to save lives, increase safety, and reduce the autonomous car crashes through research and enforcement activity. The administration of NHTSA is enforcing the safety equipment in motor vehicles and emerging Automated Driving System (ADS). However, the Automotive industry does not establish any rules and regulation to certify autonomous cars. Therefore, NHTSA has a formed a regulative body for the evaluation of new technologies and test the self-driving cars in critical scenarios before the deployment on public roads. The accidental crash report on NHTSA database does not provide any information about the car velocity, acceleration and steering angle. The crash report is semi-structured to unstructured multimedia documents with the description of the pre-crash event, collision impact of the cars and after crash event, the speed limit of the vehicles on specific road and written in natural language. The non-trivial problem in unstructured data extract meaningful information for the reconstruction of crash scenarios focusing on the aftermath of the collision such as rotation, direction and point of impact on cars.

The safety of the autonomous car unknown at design time and safe motion plan is highly unpredictable in complex or critical scenarios. It is hard to try all the possible environments in autonomous driving and autonomous vehicles will encounter the situation that have not been tested before[16]. The velocity of the car highly impacts the drivable area of each vehicle and time to reach the target point. The autonomous vehicles are look forward to adapting the velocity according to the environment and drive smoothly without any external output[17]. However, the autonomous car simulator is incapable to test the car simulators with real environments and make adjustment of the velocity for the safe driving to avoid any crash. Moreover, the speed limit on lanes, curved roads, highways make the trajectory planning high complex and computation difficult. To address this problem, the parameter space of way points and velocity will be explored for the crash scenario and how the mutant parameter affect the outcome of the pos-crash. The current reconstruction crashes from car crash report (AC3R) does not support the simulation of pos-crash car event and realistic environments to make the reconstruction of car collision simulation more accurate.

In this research, the motive is to identify the set of parameters such as speed and way points that accurately reproduce the simulation on realistic simulated environments with automatic test generation. The primary focus is the production of aftermath in the simulation as described in the police crash report. The system will apply Natural Language Processing (NLP) concepts and data mining concepts to extract the key features from semi-structured to unstructured crash reports about the vehicles, road geometry, car collision type, distance, direction and rotation. The extracted information will be used to extract the road geometry from the open street map to produced realistic simulated environment. The car crash collision will be simulated on the realistic environment to explore the velocity and way points parameters that accurately produce the aftermath of the collision. The accuracy of the collision will be evaluated based on the correct attributes of the collision of striker and victim car after the collision. The highest score of fitness function with respect to number of simulations will help us to evaluate the accuracy of the simulation.

1.1 Problem Statement

Our aim of the research is to target the aftermath information of the car crash scenarios in semi-structured to unstructured text of car crash report to generate the accurate simulation in realistic environments to enhance the accuracy of the simulation and explore the parameters of the simulation such as way points and velocity of the autonomous car to evaluate the accuracy of the simulation.

The information extraction becomes complex in the case that each word context and it relationship with other sentences has to analyzed before extracting the key feature. The aftermath events has to be determined in natural language processing text and extract all the key features such as rotation, distance/destination, direction and damage area of the car. All these features extraction has not done yet by the car crash reconstruction scenarios.

The up to date car crash simulators does not include the the realistic road geometry and environment for scenarios creation and synthetic data is used to construct the road geometry for simulations that does not reflect the true simulations from the car crash report. The realistic road geometry plays an important role in studying the real life car crash scenarios and accuracy of the simulation.

The velocity and way points parameter of the collision is difficult to address the parameter space becomes huge when considering the velocity and way points of both ego car and victim car. The autonomous cars drive on the assumption that feasible velocity reference is given and predetermined path of autonomous car and surrounding car are not going to collide with each other on given speed. The parameter space will reduce by restriction the way points according to road geometry.

1.2 Research Question

This thesis will look into the following research question:

- Explore the parameters such as way points and velocity for the reconstruction of car crash scenario of self-driving cars from police car crash report by extraction the aftermath information to evaluate the accuracy of the car crash simulation in realistic simulated environments.

2 Background and State of Art

In order to describe the reconstruction of crash scenarios of autonomous cases in detail, we will first look at some of related work to specific areas of Natural Language Processing, Open Street Map and Automatic Test Generation.

2.1 Natural Language Processing

The extraction of information extraction with the regular expression in perspective with the existing literature consist of pattern learning over tokens with various natural language processing techniques, e.g., POS tagging instead of putting effort to infer patterns over the text to be extracted [18] [19]. The generation of regex patterns for desired behavior might be different for the specified task and many approaches exist to extract key features from text [20]. The regex pattern does guarantee that a solution will perfectly fit all the scenarios in practical that actually exists. The Part of Speech (POS Tag)[21] is a technique relies on Natural Language Processing [22] to extract the key features [23] such as Noun Phrase, Verb Phrase to understand the context and meaning of the sentence.

Guarino et al [24] illustrate ontology's as "explicit specification of a shared conceptualization". The ontology's [25] consist of names, definition, properties and inter-relationship of the particular domain like automotive application and weather application. Armand et al [26] depict that how ontology's can be applied to Advanced Driving Assistance System (ADAS) to model the spatial-temporal relationship between the cars. The classified entities in ontology's to understand map, vehicle (Mobile Entity), track (Static Entity) and dynamic state (Context Parameter) of the ego vehicle. Hummel el al[27] enlightened the description logic to find the intersection and geometry of roads. Similar approaches to scene understanding for ADAS by Zhao et al [28] to obtain information about traffic, vehicle and and open street map.

2.2 Open Street Map (OSM)

Godoy et al[29] come up with procedure to identify the road intersections and junctions in OpenStreetMap [30]. The path were extracted from the Open Street Map based on the key attribute such as road type, surrounding area and scenic beauty. Guillaume et al [31] surveyed the challenges in Simultaneous Localization And Mapping (SLAM) for the localization of driving cars in Enhanced Maps to resist the environment variability (weather, season). The paper provides an overview of the challenges in large scale experiments of autonomous cars. The synthetic data for simulated environment for simulation still has a huge gap with realistic simulated environments [32]. In order to solve this issue, the open source open street map provide the geo data for the whole world to generate maps of the whole city[14].

2.3 Automatic Test Case Generation

In Erbsmehl [33], the author focus on the simulation of advanced safety systems on real world crashes based on German In-Depth Accident Study database. The comparative and analysis study is done on simulated real world accident scenario and a predicted accident scenario on virtual prototype of safety system to estimate the benefit of safety system in vehicles. The GIDAS pre crash matrix contains all the relevant information such as the geometrical information of the accident (driving lane of each participants, lane borders, lane markers, view obstacles) and the dynamic behaviour of the participants. CarSim with Matlab is used for the simulation each participant and vehicle dynamic with additional car parameter, physical parameters is calculated at every 0.001 seconds. Squillant et al[34] focus on the modeling of accident scenarios with missing data in the databases and the design of safety systems related to the accident. Zhou et al[35], the author proposes a Gaussian Process Classification algorithm to efficiently determine the non-convex boundaries based on the limited number of simulations. The modelling of the safety boundary search consist of a test case "yaw" with 3 dimensional inputs such as longitudinal inter-vehicle distance,the longitudinal velocity difference between ego vehicle and cut-in vehicle, cut-in angle yaw of the cut-in vehicle and and binary outputs (Collision or not Collision).

Arrieta et al[36] studied the traditional testing techniques, Model based testing are not feasible to apply to Cyber Physical Systems because it is not feasible to evaluate all the possible state and values, So the simulation models with search-based approach are developing to execute large number of reactive test cases with prioritization to effectively execute test cases with critical scenarios and replication of safety critical functions. The fitness function with four objectives functions are analyzed with crossover and mutation operator incorporating five multi-objective search algorithms is evaluated in case studies to find the optimal solution. Furthermore, the modules of 3D simulation to demonstrate the Scene Graph Model and validate the trajectory and autonomous component. The number of traffic scenarios can be evaluated by model parameterization to increase the coverage of testing space[37].

The AC3R, reproduced the car crash scenarios from the police reports by automatic natural language to scene creation for computer generating simulations. The simulations were accurate enough to cover all the pre-crash event, but the pos-crash event of collision still needs to be done. The diversity of test scenarios in virtual testing expose the critical bug in the software[38].

3 Proposed Method

The aim of the research is to explore the parameter crash scenario such as waypoint and speed to produce the accurate simulation of autonomous cars in realistic environments. The velocity parameter is not given on the police crash report; thus, it makes the aftermath analysis of car crash make more difficult. So, our primary focus is to get the numerical values from the simulation such as direction, impact or damage area of the car, angle of rotation and destination lane. In the end, out fitness function will help us to evaluate the accuracy of the simulation based on the extracted numerical values from the simulation.

In short, the approach is to apply natural language to extract key features of the aftermath from the semi-structured to unstructured text and combined with pre-crash event features of the AC3R to create the simulation. The data extracted contain all the information of road geometry, the direction of the car, point of impact of a car crash, car rotation and destination lane after impact. The road geometry will us to extract the roads from the Open Street Map as describe in the crash report to create a realistic environment to improve the accuracy of the simulation. After the creation of the scenario on BeamNG, the striker and victim car will be simulated with different speed and waypoints for the collision of vehicles. The number of simulations will be run to get the numerical values of the parameter of the critical crash scenario until the stop criterion is reached. On each simulation, the direction, impact or damage area of the car, angle of rotation and destination lane of each car will be evaluated with fitness function to determine the accuracy of the simulation.

The procedure of automated test cases generation to explore the parameters of the car accident scenario is summarized in the Figure 2.

3.1 Natural Language Processing

In the growing world of internet and online services, the data is available in gigabytes in open source platforms and communicated in the form of text. The extracted text is available in the form of tabular data, unstructured data, and raw data in XML. Our target is to apply the Natural Processing algorithms to understand the human language that what are they saying about the environment of the car crash scenario, collision, and aftermath of autonomous cars.

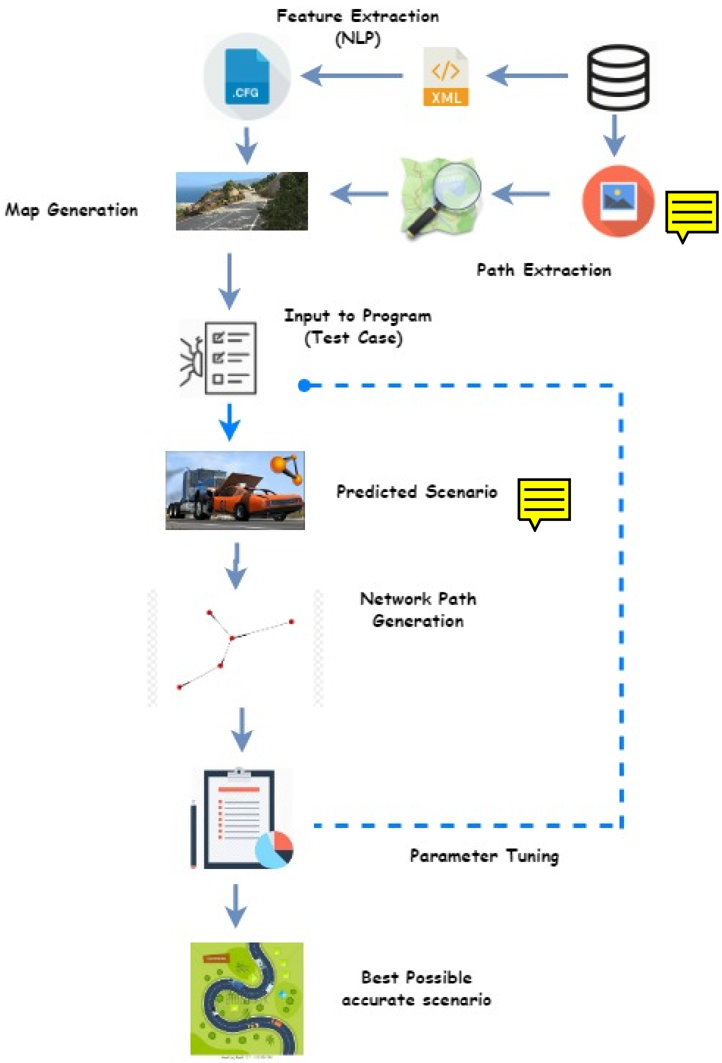

Figure 1: Architecture Diagram

3.1.1 Text Analysis and Features Extraction

To generate the trajectory based on the car crash report, the following features need to be extracted using Natural language processing Algorithms.

- Initial direction of striker and victim car (Pre-crash Event)

- Road Geometry (Pre-crash Event)
- Point of Impact/Collision (Crash Event)
- Rotation of striker and victim car (Pos-crash Event)
- Car facing direction after impact (Pos-crash Event)
- Destination Lane after impact (Pos-crash Event)

All these features extraction is based on the information available in the semi-structured and unstructured data in the crash report. If the information is not available, the simulation will not be generated. The pre-crash event information will not extracted from police crash summary in this but it will use the pre-crash and crash event extraction of AC3R.

3.1.2 Natural Language Processing Algorithm

These individual steps are required to convert text into the machine-readable format for further processing. The regex equations, nltk and text mining should be used to execute all the steps.

All the steps defined below is based on the XML from MVCCS database from NHTSA. The XML can be downlaoded from the given link : caseid=2007005289582

Summary extraction from XML - The initial step is to extract the crash summary, number of vehicles and crash report id from the XML. ~~The file name can be string and return a document as an XML type. The crash summary can be get by function *find()*, which return the elements that matches the specified criteria~~. The target node in XML are

- CaseID
- NumOfVehicle
- Crash/CRASH/XML_CASESUMMARY/SUMMARY

All the information extraction is stored in the json that is required for further processing of features extraction from xml. The summary extracted from the XML is given below.

> This two-vehicle crash occurred in the early evening hours on a weekday, in the middle of a T-intersection. The two intersecting roadways were both two-lane undivided bituminous roadways that were dry, level and straight at the crash location. The westbound roadway ended at the intersection and was controlled by a stop sign. There was no control device for north or southbound traffic and the posted speed limit was 56 kmph (35 mph). There were no adverse atmospheric conditions at the time of the crash and traffic flow was normal. It was dark out and the roadway was not illuminated. Vehicle one (V1), a 2003 Honda Accord 4-door sedan, was heading west on the roadway approaching the intersection. Vehicle two (V2), a 1982 Dodge Ram van, was heading north on the roadway intending to continue straight through the intersection. As V1 entered the intersection, its left side was struck by the front of V2, which had depressed its brakes, leaving approximately 40 meters of skid marks. After the initial impact, V1 rotated counterclockwise approximately 130 degrees before coming to final rest partially off of the roadway at the top of the intersection facing

east. Vehicle two rotated clockwise approximately 45 degrees and continued off the right side of the roadway where it came to final rest. Both vehicles had to be towed from the scene due to damage. The driver of V1, a 68 year-old male, had to be extricated from his vehicle and was transported to a medical facility for treatment. According to the medical reports, he suffered two mild fractures of the spine at location C6. There were no other occupants in the vehicle when the crash occurred. After numerous attempts, an interview could not be obtained from the driver of this vehicle. For this reason, it is uncertain whether any fatigue or health related factors contributed to this crash. The Critical Pre-crash Event coded to this vehicle was: this vehicle traveling in an unknown direction. The reason for this coding was because the driver's intentions were unclear without an interview. The Critical Reason was coded as: unknown driver related error. Associated factors coded to the driver of this vehicle include the use of prescription medication to control diabetes and failure to obey a traffic control device (stop sign). The driver of V2, a 55 year-old male, was not injured as a result of the crash and did not require any medical treatment. There were no other occupants in the vehicle when the crash occurred. A complete interview was obtained from the driver of this vehicle. He stated that he saw the other vehicle at the intersection and couldn't believe that the guy was actually going to go. He then claimed that V1 stopped in the middle of the intersection and that he could see the driver's eyes. The driver of V1 relayed that it was like witnessing a "deer in headlights." As soon as he realized things were not right, he stood up on the brakes and tried to steer to the right in an attempt to avoid the vehicle that was in the intersection, but he could not avoid it. The driver related that he was not tired and that he was familiar with the roadway. He stated that he was in good health and that he had not taken any medication, prescription or otherwise, prior to the crash. The driver was very comfortable with his vehicle and there were no mechanical contributing factors. The Critical Pre-crash Event coded to this vehicle was: other vehicle encroachment from crossing street, intended path unknown. The Critical Reason was not coded to this vehicle. No associated factors were coded to the driver of this vehicle.

Pre-processing Summary - The initial step of analyzing the text content of summary is to keep all only alphanumeric characters in the summary and remove all the tabs, extra spaces and newlines. Secondly, the different identifying names of each vehicles of striker and victim is reduced to single identifier.

'v1', 'vehicleone', 'vehicle one', 'vehicle 1', 'Vehicle 1' $\Rightarrow' V1'$.

'v2', 'vehicletwo', 'vehicle two', 'vehicle 2', 'Vehicle 2' $\Rightarrow' V2'$.

Pos-Crash Event Extraction - After preprocessing of the summary, the next goal is to identify pos-crash event in the summary. The pos crash event is often describe with the crash event. The specific crash and after impact keywords based on the contextual information is investigated the sentences. All the other information is removed while extraction the aftermath features. The pos-crash extracted sentences from the crash report sample is given below.

V1 (V1), a 2003 Honda Accord 4-door sedan, was heading west on the roadway approaching the intersection. V2 (V2), a 1982 Dodge Ram van, was heading north on the roadway intending to continue straight through the intersection. As V1 entered the intersection, its left side was struck by the front of V2, which had depressed its brakes, leaving approximately 40 meters of skid marks. After the initial impact, V1 rotated counterclockwise approximately 130 degrees before coming to final rest partially off of the roadway at the top of the intersection facing east. V2 rotated clockwise approximately 45 degrees and continued off the right side of the roadway where it came to final rest. Both vehicles had to be towed from the scene due to damage.

Stops Word Removal - In sentence, there are unnecessary words/token that don't carry any meaning and does not determine and semantic relation between the words are called ”Stops Word”. The words like “the”, “a”, “on”, “is”, “all” are usually removed from the text to carry out further processing. The stop word processing will be applied after finding the specific keyword in the text.

Dictionary Expansion - Since, I have to look for the specific keywords in the sentences to extract rotation, direction and destination. The vocabulary in the NHTSA database is not restricted. Hence, I have the expand the keywords set with Google News word2vec to increase the accuracy in features extraction. The word2vec take the keyword as input and produces the word of vector as output. It is useful in getting the possible representation of words in text. The most common keyword for direction extraction is ”facing”. I expanded it with GoogleNews word2vec to get all the possible combination of the words.

facing $\Rightarrow faced, faces, face, confronting, Facing$

Rotation Extraction - In the pos-crash event, the sentence with the rotation keyword, expanded dictionary is used to extract features of the rotation of the vehicles. The rotation of the car describes the rotation of the striker and victim car from the point of impact. Clockwise and Counter-clockwise rotation are two type rotation that are describe in the crash report. The clockwise rotation is that car rotate toward right after the impact with another car and counterclockwise rotation is the rotated towards the left after the impact. The reference point is the direction of the vehicle just before the impact. The identified sentence for the rotation based on the 'rotated' keyword is

After the initial impact, V1 rotated counterclockwise approximately 130 degrees before coming to final rest partially off of the roadway at the top of the inter-section facing east. V2 rotated clockwise approximately 45 degrees and continued off the right side of the roadway where it came to final rest.

After the identification of rotation context sentences, the clockwise and counter clockwise word is identified and the sub string is extracted with the help of regex patterns.

Step 1) regex pattern for pre-word sentence = r'.+?(?=' + word + ')' Step 2) Tokenization and reversed

The end result of the vehicle one counterclockwise direction and vehicle two clockwise direction.

Degrees Extraction - In the pos-crash event, the sentence with the degrees keyword, expanded keyword dictionary is used to extract features of the degree of rotation of the vehicles. The degree of rotation of each car describes the rotation of the striker and victim car from the point of impact. The degree of rotation should be Clockwise and Counter-clockwise rotation in the crash report. The clockwise rotation is that car rotate toward right after the impact with another car and counterclockwise rotation is the rotated towards the left after the impact. The reference point is the direction of the vehicle just before the impact.

After the initial impact, V1 rotated counterclockwise approximately130 degrees before coming to final rest partially off of the roadway at the top of the inter-section facing east. V2 rotated clockwise approximately 45 degrees and continued off the right side of the roadway where it came to final rest.

After the identification of degrees in sentences, the number of degrees counterclockwise and the number clockwise word is identified and the sub string is extracted with the help of regex patterns.

Step 1) regex pattern for pre-word sentence = r'.+?(?=' + word + ')' Step 2) Tokenization and reversed

The end result of the vehicle one in degrees rotation is 130 and vehicle two in degrees rotation is 45.

Direction Extraction In the pos-crash event, the sentence with the facing keyword, expanded facing dictionary is used to extract features of the direction of vehicle when it came to rest after impact. The reference point of the direction is reference to north, west, south and east direction of the scenario.

After the initial impact, V1 rotated counterclockwise approximately130 degrees before coming to final rest partially off of the roadway at the top of the inter-section facing east.

After the identification of facing context sentences, the direction of the facing is identified with pos-word substring (sentence) and the subject with pre-word substring (sentence).

Step 1) regex pattern for pre-word sentence = r'.+?(?=' + word + ')' Step 2) regex pattern for pos-word sentence = r'(?<=' + verb + ').*' Step 3) Tokenization and reversed

The end result of the vehicle one is facing east and vehicle two is considered as 'initial direction' or 'default direction' as no facing information is giving about vehicle two.

Destination lane Extraction - The Destination lane is how far does the car goes from the point of impact. There is no information about the distance of each striker and victim car from crash point. Secondly, each crash report has different way of describing the car behavior from crash point. For instance, the crash car ran off the road, the car end in westbound lane, the car strikes the signal, the car become rest at intersection. It is impossible to cover all the scenarios. Hence, we have to compromise between the car destination point and extracting all the information from the crash report. In our simulation, we have divided the destination lane into two categories such as "on road" and "off road". The "on road" means that car staying within the bounding box or stay on the road after the point of impact and "off road" means that the car goes outside the bounding box or lay off the road after point of impact.

After the identification of "on road" and "off road" context sentences, the sentences are evaluated to determine the position of each vehicle from the point of impact. It is not possible to cover all the scenarios, hence the explanation of most similar scenarios in crash report is not valid. The generalize approach of "on road" and "off road" is develop to avoid over-fitting of crash scenarios and cover most of the crash scenarios in the best possible way.

> V1 rotated counterclockwise approximately 130 degrees before coming to final rest partially off of the roadway at the top of the intersection facing east. V2 rotated clockwise approximately 45 degrees and continued off the right side of the roadway where it came to final rest.

The end result of the vehicle one stays "on road" and vehicle two goes "off the road".

All the information extracted ~~is stored in the XML/JSON file or csv file, it~~ will be further processed to create scenarios in BeamNG by extracting road Geometry from Open Street Map.

3.2 Open Street Map

The Open Street Map (OSM) is an open source platform to extract the data about roads and nearby characteristic such as building in the given region of interest. Open Street Map allow users to edit map and use data free of cost. Our objective is to extract the data from the open street map according to the description given in the natural language processing. The roads can highways, urban road, rural road, junctions. However, the location is rarely specified in the description so we will assume our region of interest and try to extract the best possible scenario from Open Street Map.

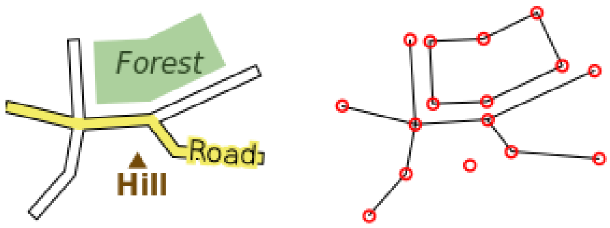

Figure 2: Open Street Map Nodes

3.2.1 Data Extraction from Open Street Map

The Open Street Map[30] data can be extracted from Xapi Api. After obtaining the response from API request, the XML object is parsed to extract the roads and link roads according to the tagged provided in the smart query. The target attribute is the width of the road and it can be filtered from large data-set and small scenario can be drawn on the BeamNG.

OpenStreetMap is an opensource data manage by the open street map community and the people to gather GPS data of roads, building arounds the world. Open Street Map has a simple and flexible data model that make it easier for researchers to extract the required data from it. The Open Street Map has following data primitive types.

1. **Nodes** have an id and a location, given in longitude and latitude coordinates.
2. **Ways** consist of an ordered list of nodes. Depending on its context, it may represent a line (e.g. a street) or an area..

The data model is XML/JSON contains all the semantic information encoded in in tags, key-value pairs. The data model tags are standardized and defined by open street map community to represent the commonly use information. Here is the list of common tags will be used to extract the required information.

Table 1: Commonly used tags in OpenStreetMap and their meanings

Tag	**Meaning**
Highway	The Type of road
Lanes	Number of lanes on each road
Width	The width of the road in meters
signal	Signal at the intersection/junction

All the steps defined below for procsing of open street map data is based on the XML from MVCCS/NASS database from NHTSA.

Extraction of Roads from Overpass API - The open street raw xml contains the information about roads, building, pathway and street etc. To avoid heavy preprocessing of data and reduced the errors in data. We extract the data from the Overpass API that returns only the roads of the given location. The sample query is

```
overpass_query = [out:xml];(
way
(around:100,48.571847,13.4607844)
[highway~"^(primary|secondary|tertiary|residential)$"]
[name];>;);out;
```

Extraction of Road Geometry - The required open street map data is extracted from the overpass API. Further processing is applied on the raw data to extract the road geometry of each node. Usually, all the nodes and ways are distributed over open street map raw data. The open street map data is processed according to our requirement and stored in the elastic search. The following attribute of each node (latitude and longitude) is saved in the elastic that is required for scenario creation in BeamNG. The attributes are number of lanes, width, side walk, max speed, lane backward, lane forward, road sign. For intersection or junctions, the attributes are highway, crossing and road sign.

Migrate all openstreetmap XML data to Graph using networkx - Connected Graph Since, we need the network of road map around the city to move the vehicles on real world simulation environment. So, we migrate all the data of open street map to Graph. All the information of nodes is mapped to graph where each node is latitude and longitude. The edge between each nodes represent the connecting roads. The weight of each node is the distance between the two nodes. Finally, we perform a reachability check of all roads and remove those, that cannot be reached from the main part of the map, thus eliminating small isolated clusters, that might have been introduced as a map only contains a rectangular section of the world.

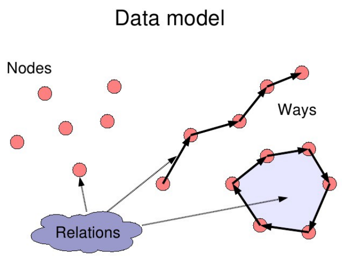

Figure 3: Open Street Map Nodes and Ways relation

Traverse Graph for Intersection point - As mentioned earlier, the Graph is the set connected nodes that represent the road network of the real-world network. Our target is to find the roads on the graph according to description of the police crash report. The user has to specify the starting position on the Graph or the random node will be picked up as a starting point. Now our routing algorithm will find any route that according to the required the description and we need to search only those nodes which are guaranteed to be connected. The non-trivial task is to find the T-intersection (3-way intersection) and intersection (4-way intersection) in the connected Graph. Our routing algorithm check for the following conditions to find the 3-way intersection and 4-way intersection.

1. **3-way** is an intersection has a degree 3, which is the junction node connected to 3 different node.
2. **4-way** is an intersection has a degree 4, which is the junction node connected to 4 different node.

In the NHTSA sample, the police crash report describe the type of intersection where the collision of vehicles occurred.

This two-vehicle crash occurred in the early evening hours on a weekday, in the middle ofa T-intersection, The extracted sentence for the type of intersection is

The Graph is traversed to find the 3-way intersection with undivided road. The coordinates (latitude and longitude) of the required road is extracted. The output of the road is shown on mp-leaflet.

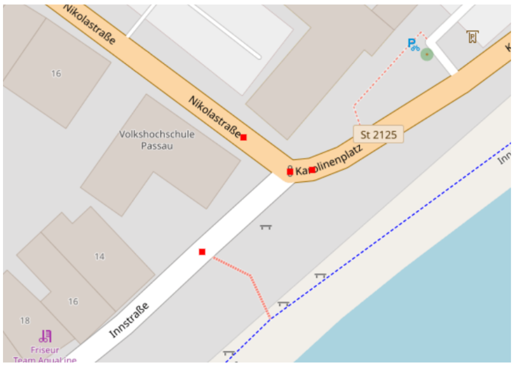

Figure 4: T-Intersection Open Street Map

The standard width of the single lane on road in open street map is 4 meter. The number of lanes extracted of each road extracted from the open street map data is

(node, coordinates, number of lanes, width)
('595019', (48.571847, 13.4607844), '2', 0) - Junction Point.
('3278295187', (48.571847, 13.4607844), '3', 0)
('60775846', (48.571847, 13.4607844), '2', 0)
('1853222750', (48.571847, 13.4607844), '2', 0)

3.3 Environment Reconstruction - Open Street Map Scenario Creation

The simulation of self-driving car crash scenario requires a realistic environment instead of simulation environment created on synthetic data based on assumptions. The BeamNG research using Python provide a Grid Map which enables you to create a custom trajectory map for cars. The road extracted from open street map will be reconstructed on BeamNG according to the description of the police crash report. It is necessary that the assumption on BeamNG map creation actually reflect the original scenario describe in Vehicle Crash report.
The BeamNG required following parameters as input for scenario creation

- Direction of road.
- Number of lanes
- Length of each road.

- Angle between connected roads.

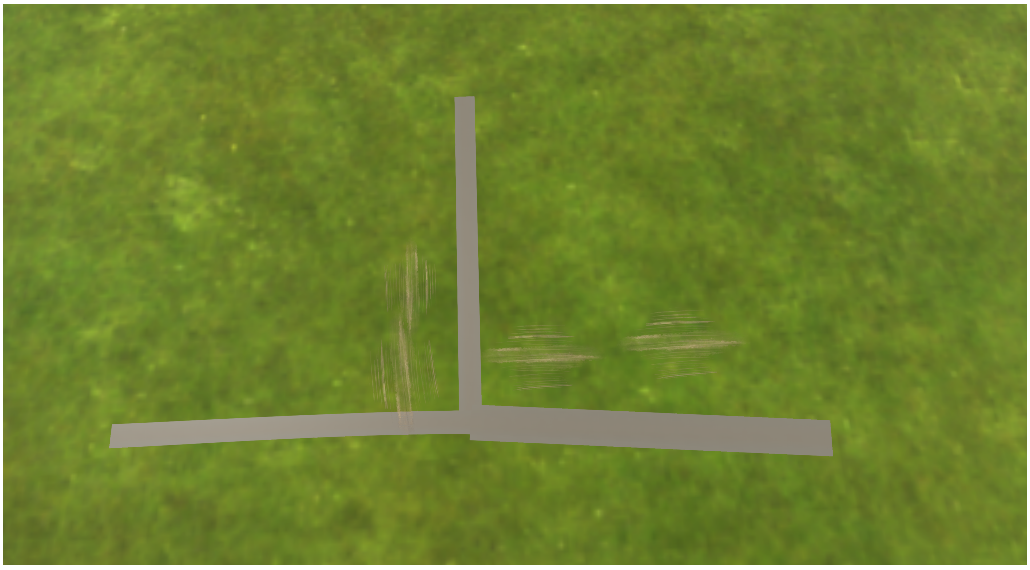

Figure 5: BeamNG Scenario

3.4 Generation Test Cases Scenarios in BeamNG

Simulations experiments are the fundamental tool to study and develop vehicle accidents of self-driving cars with an integrated option of Artificial Intelligence. The BeamNG research provide the realistic figures of car velocity, car dynamics, car damage and vehicle state at each timestamp. It helps us to predict future events, carry out analysis on extracted data. All the simulations on police crash report is based on the assumption that striker vehicle and victim vehicle should be in free state (Steering= 0, acceleration= 0, braking=0) before collision. There are two ways to automatically generate the test cases.

Parameter Exploration The visiting of those regions in the parameter space that is not explored or visited yet.

Parameter Exploitation means that given a reasonable solution, the algorithm will keep refining that solution until it reaches a local optimum.

3.4.1 Parameter Exploration (Oracle)

As mentioned, our objective is to explore the parameters for self-driving cars by executing the number of test cases that reflect the original scenario. Since, the region space is quite large and contained the large number of values. Hence, it will increase the simulation time of the test case of the given scenario until we find an optimal solution. One way to explore the parameters is visiting of those regions in the parameter space that is not explored or visited yet.

Our target is to produce the accurate pos-crash event according to the natural language description in police crash report. The accurate pos-crash event of car requires the collision of striker vehicle and victim vehicle at different speed and waypoints. The exploration of optimal parameter heavily relies on the Aftermath of road accident. The Aftermath analysis of the accident and accuracy of the simulation will help us to choose the next parameters for the test cases to produce simulation nearest to the ground truth. Initially, the simulation will be produced with given set of random parameters and grid search to calculate the fitness of each simulation. After getting the set of results, the user (orcale) will choose the next parameters to get the accurate pos-event.

3.4.2 Vehicle State Extraction From BeamNG

The BeamNG provide the feature to extract the key components such as speed, position, direction and damage of the car in the simulated scenario. During the simulation of the car crash scenario with different parameters, the velocity, position, direction and rotation of the car be extracted during the simulation to evaluate the accuracy of the pos-crash event in simulation.

In addition, the damage of the striker vehicle and the victim vehicle will be considered to know the exact point of impact on the roads and which region of the car is damaged. It will help us to gather critical (collision of cars) scenarios and parameters in the large parameter space to narrow down the simulation time. Moreover, the numerical values will be monitored at every 0.5 sec to 1 sec to reduce the computation times and reduced the number of nodes on the graph. The NetworkX [39] library enable us to convert the continuous data into discrete nodes.

3.4.3 Fitness Function (Accuracy)

The simulation accuracy will be evaluated on the number of correct attributes in the post-crash event of the police car crash simulation. The score accuracy will be calculated for each simulation and the next parameter will be chosen based on the current fitness score. We can restrict the regions of the search space based on the feedback from the simulation and reduce the time of simulation and computation cost. Since, our fitness function can be run on the infinite solutions to achieve the desired score. Therefore, the number of simulations will be run for 1 hour and the accuracy of the simulation will be evaluated. The simulation with best fitness function will be our final output.

Striker Vehicle					
	damage_area	**rotation**	**degrees**	**facing**	**destination_lane**
True Value	Left side	clockwise	130	east	on road
Predicted Value	Left Side	clockwise	100	east	off road
Correct	yes	yes	yes	yes	no

Victim Vehicle					
	damage_area	**rotation**	**degrees**	**facing**	**destination_lane**
True Value	Front	counterclockwise	45	initial	on road
Predicted Value	Front	clockwise	45	opposite	on road
Correct	yes	no	yes	no	yes

Fitness Function	
Fitness Formula	Number of Correct Output x 0.125
Fitness Score	(7 x 0.10) = 0.7
Fitness Threshold	0.70

if the fitness equation value is equal to fitness threshold score or more than the threshold. It means that pos-crash event simulation is accurate enough for the accepted result. The fitness function will be applied to each evaluation and our goal is to maximize the fitness function for accurate pos-crash reproduction.

3.4.4 Parameter Exploitation (Automating Test Cases)

In parameter exploration, the user has to choose the parameter himself to explore the optimal parameter. However, parameter exploitation will generate the next parameter with the help of genetic evolution algorithm. Parameter exploitation is based on the genetic algorithm to select and produce the fittest individuals to produce the offspring for the next generation. Hence, the selection of optimal parameter is the challenging task to produce accurate pos-crash event simulation. The figure below shows the flow of the genetic algorithm.

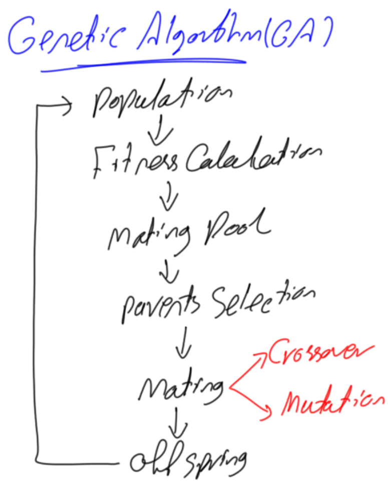

Figure 6: Genetic Algorithm

Chromosome Representation The representation of the chromosome is the problem specific and representation should make search space smaller and easier to search. Hence, our chromosome representation should be binary string string between zeros and ones and float range of values between zeros and ones. In our case, the real value encoded chromosome for automatic test generation is shown on the table below.

Real Value Encoded Chromosome					
Striker Speed	**Striker X1**	**Striker Y1**	**Victim Speed**	**Victim X1**	**Victim Y1**
35 m/s	-60	-1	25 m/s	2	40

The generation of random population between the given range of numbers becomes difficult. Hence, we must have to convert our real value encoded chromosome to binary chromosome to make processing easier. Furthermore, the binary encoded chromosome will help us in crossover and mutation to generate new off-spring. The value of each gene lies between 0 and 1.

Binary Encoded Chromosome					
Striker Speed	**Striker X1**	**Striker Y1**	**Victim Speed**	**Victim X1**	**Victim Y1**
0.35	0.60	0.25	0.25	0.50	0.40

Population Size Population is the set of chromosomes that are created randomly, The diversity of the should be large enough to avoid premature convergence. The population size should be optimal to avoid slowing down of the Genetic Algorithm and select good chromosomes for mating pool. The fitness of each chromosome is calculated to find the potential candidate for selection in Genetic Algorithm and keep the elites for future generation.

Mating Pool - Tournament Selection Tournament selection involves multiple tournament of the chromosome (chromosome) individuals from the random selection. The fittest candidate in the tournament selection is selected for crossover and mutation. The tournament size should be smaller because it has high probability to get all individuals selected and preserves diversity. Another design feature to consider is the use of elitism. With elitism, the best performing individuals from the population will automatically carry over to the next generation, ensuring that the most successful individuals persist.

Crossover and Mutation The next generation to explore the good individuals, we perform the crossover to create the next generation. The crossover point is simply picked to splice a two string and produce an offspring. However, mutation serves to avoid local convergence by adding noise (small numerical value) in the offspring generation. The mutation is applied to gene of the offspring that are subject to the mutation with a low random probability.

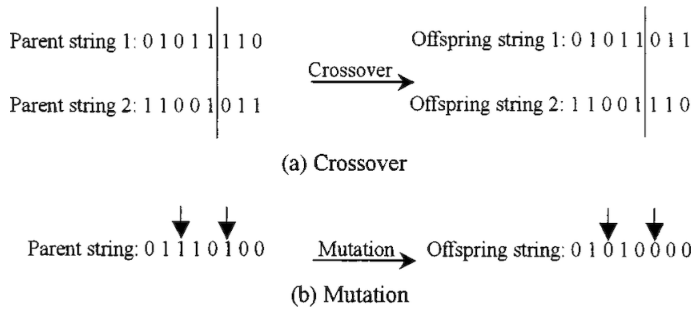

Figure 7: Crossover and Mutation

Selected Parents and Offspring Evaluation (Survivor Selection) In survivor selection for the new population, the finite individuals are allowed to reproduce the offspring and kicked out of the population after evaluation of the fitness of parent and offspring.

Fitness Function The simulation accuracy will be evaluated on the number of correct attributes in the post-crash event of the police car crash simulation. The score accuracy will be calculated for each simulation and the next parameter will be chosen based on the current fitness score. We can restrict the regions of the search space based on the feedback from the simulation and reduce the time of simulation and computation cost. Since, our fitness function can be run on the infinite solutions to achieve the desired score. Therefore, the number of simulations will be run for 1 hour and the accuracy of the simulation will be evaluated. The simulation with best fitness function will be our final output.

Striker Vehicle					
	damage_area	**rotation**	**degrees**	**facing**	**destination_lane**
True Value	Left side	clockwise	130	east	on road
Predicted Value	Left Side	clockwise	100	east	off road
Correct	yes	yes	yes	yes	no

Victim Vehicle					
	damage_area	**rotation**	**degrees**	**facing**	**destination_lane**
True Value	Front	counterclockwise	45	initial	on road
Predicted Value	Front	clockwise	45	opposite	on road
Correct	yes	no	yes	no	yes

Fitness Function	
Fitness Formula	Number of Correct Output x 0.125
Fitness Score	(7 x 0.10) = 0.7
Fitness Threshold	0.70

if the fitness equation value is equal to fitness threshold score or more than the threshold. It means that pos-crash event simulation is accurate enough for the accepted result. The fitness function will be applied to each evaluation and our goal is to maximize the fitness function for accurate pos-crash reproduction.

Termination The algorithm terminates if the population has converged, thus the genetic algorithm has provided a set of solutions to our problem.

4 Planned Evaluation

Our aim of the research to explore the parameters such as way points and velocity for the accurate reconstruction of car of pos-crash event in car crash scenario of self-driving cars from police crash report by extracting the aftermath information to evaluate the accuracy of the car crash simulation in realistic simulated environment. All the parameters of pos-crash event and realistic environment plays a vital role in determining the accuracy of reconstructed car crash scenario. The results are bases on the assumption that car mass, inertia is same for all autonomous cars in the crash report. The car type must be remain same for all the simulations of the parameter exploration and exploitation.

A lot of simulation should be run to get the required parameters and aftermath close to the original scenario. Since, the number of simulations is not defined, we will restrict the number of simulations for 1 hour for both parameter exploration and parameter exploitation and compare the fitness function values of each simulation. The target is to achieve the maximum fitness value of the pos-crash event in car crash simulation.

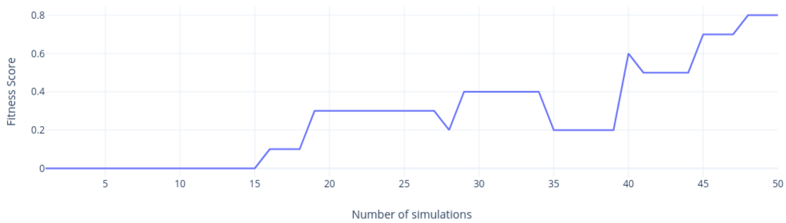

Figure 8: Parameter Exploration Score

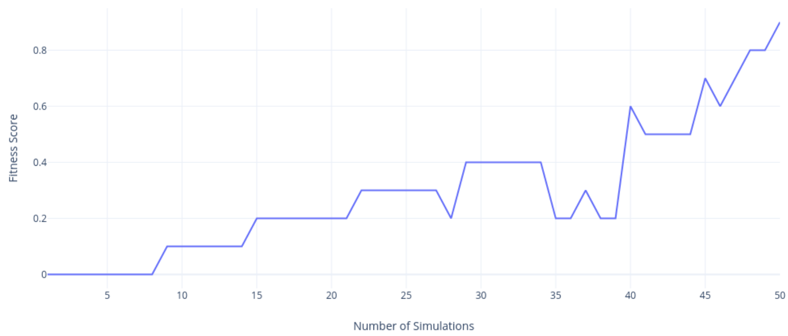

Figure 9: Parameter Exploitation Score

The top N fitness simulation will be selected from the series of simulation to get the best accurate result as close to the description of natural language. The parameters that produce the highest fitness score will be our final output.

Final Output					
Striker Speed	**Striker X1**	**Striker Y1**	**Victim Speed**	**Victim X1**	**Victim Y1**
35 m/s	-60	-1	25 m/s	2	40

5 Schedule

I would like to start my work on thesis in September 2019 and follow up-to next 6 months. The milestone of each task and time duration is summarized in given schedule.

Start Date : 1st September 2019
End Date : 1st March 2020

The schedule of my 6 months is divided into 24 weeks and given in the Table 1.

6 Success criteria

The research work of my master thesis should contain the following features to evaluate the end results and completion of all these features to be considered as completed or successful.

Table 2: Thesis Schedule

Weeks	Task
4	Understanding NHTSA Accident Summary and Data Extraction
4	Open Street Map Data Extraction
4	BeamNG scenario creation and Vehicle state Extraction
4	Generation Test Case for Parameter Exploration and Parameter Exploitation
2	Simulation of Police Crash for Result Evaluation
6	Thesis Writing

Table 3: Summary of the Expected Thesis Features

Feature	Must-Have	May-Have	Must-Not Have
Crash Report Data Extraction (Aftermath)	✓	-	-
Open Street Map Data Extraction	✓	-	-
Scenario Creation in BeamNG (Road Geometry)	✓	-	-
Parameter Exploration	✓	-	-
Parameter Exploitation	-	✓	-
Fitness vs (Time or Number of Simulations)	✓	-	-
Trajectory Path Generation (Polyline)	✓	-	-
Minimum 3 Crash Report	✓	-	-
Artificial Intelligence support for BeamNG	-	✓	-

References

[1] PA Hancock, Illah Nourbakhsh, and Jack Stewart. On the future of transportation in an era of automated and autonomous vehicles. *Proceedings of the National Academy of Sciences*, page 201805770, 2019.

[2] Michele Bertoncello and Dominik Wee. Ten ways autonomous driving could redefine the automotive world. *McKinsey & Company*, 6, 2015.

[3] Tasha Keeney. Mobility-as-a-service: Why self-driving cars could change everything. *Ark Invest*, 1(3), 2017.

[4] Eric R Teoh and David G Kidd. Rage against the machine? google's self-driving cars versus human drivers. *Journal of safety research*, 63:57–60, 2017.

[5] Jinwei Zhou and Luigi del Re. Identification of critical cases of adas safety by fot based parameterization of a catalogue. In *2017 11th Asian Control Conference (ASCC)*, pages 453–458. IEEE, 2017.

[6] Nidhi Kalra and Susan M Paddock. Driving to safety: How many miles of driving would it take to demonstrate autonomous vehicle reliability? *Transportation Research Part A: Policy and Practice*, 94:182–193, 2016.

[7] Sam Levin. Uber crash shows' catastrophic failure'of self-driving technology, experts say. *The Guardian*, 22, 2018.

[8] Puneet Kohli and Anjali Chadha. Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash. In *Future of Information and Communication Conference*, pages 261–279. Springer, 2019.

[9] H Muir. Google self-driving car caught on video colliding with bus. *The Guardian https://www. theguardian. com/technology/2016/mar/09/google-self-driving-car-crash-video-acci dent-bus*, 2016.

[10] Chi Zhang, Yuehu Liu, Danchen Zhao, and Yuanqi Su. Roadview: A traffic scene simulator for autonomous vehicle simulation testing. In *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 1160–1165. IEEE, 2014.

[11] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: Automated testing of deep-neural-network-driven autonomous cars. In *Proceedings of the 40th international conference on software engineering*, pages 303–314. ACM, 2018.

[12] Raja Ben Abdessalem, Shiva Nejati, Lionel C Briand, and Thomas Stifter. Testing advanced driver assistance systems using multi-objective search and neural networks. In *Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering*, pages 63–74.

[13] Christoph Sippl, Florian Bock, David Wittmann, Harald Altinger, and Reinhard German. From simulation data to test cases for fully automated driving and adas. In *IFIP International Conference on Testing Software and Systems*, pages 191–206. Springer, 2016.

[14] Moritz Göbelbecker, Christian Dornhege, et al. Realistic cities in simulated environments-an open street map to robocup rescue converter. In *Fourth International Workshop on Synthetic Simulation and Robotics to Mitigate Earthquake Disaster (SRMED 2009)*, 2009.

[15] Adam Stocker and Susan Shaheen. Shared automated vehicle (sav) pilots and automated vehicle policy in the us: Current and future developments. In *Road Vehicle Automation 5*, pages 131–147. Springer, 2019.

[16] Matthias Althoff and Sebastian Lutz. Automatic generation of safety-critical test scenarios for collision avoidance of road vehicles. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1326–1333. IEEE, 2018.

[17] Florent Altché, Philip Polack, and Arnaud de La Fortelle. High-speed trajectory planning for autonomous vehicles using a simple dynamic model. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1–7. IEEE, 2017.

[18] Dr Ciravegna et al. Adaptive information extraction from text by rule induction and generalisation. 2001.

[19] Tianhao Wu and William M Pottenger. A semi-supervised active learning algorithm for information extraction from textual data. *Journal of the American Society for Information Science and Technology*, 56(3):258–271, 2005.

[20] Robert A Cochran, Loris D'Antoni, Benjamin Livshits, David Molnar, and Margus Veanes. Program boosting: Program synthesis via crowd-sourcing. *ACM SIGPLAN Notices*, 50(1):677–688, 2015.

[21] Lluís Màrquez and Horacio Rodríguez. Part-of-speech tagging using decision trees. In *European Conference on Machine Learning*, pages 25–36. Springer, 1998.

[22] Edward Loper and Steven Bird. Nltk: the natural language toolkit. *arXiv preprint cs/0205028*, 2002.

[23] Ruslan Mitkov, Richard Evans, Constantin Orăsan, Viktor Pekar, et al. Anaphora resolution: To what extent does it help nlp applications? In *Discourse Anaphora and Anaphor Resolution Colloquium*, pages 179–190. Springer, 2007.

[24] Nicola Guarino, Daniel Oberle, and Steffen Staab. What is an ontology? In *Handbook on ontologies*, pages 1–17. Springer, 2009.

[25] Rudi Studer, V Richard Benjamins, and Dieter Fensel. Knowledge engineering: principles and methods. *Data & knowledge engineering*, 25(1-2):161–197, 1998.

[26] Alexandre Armand, David Filliat, and Javier Ibañez-Guzman. Ontology-based context awareness for driving assistance systems. In *2014 IEEE intelligent vehicles symposium proceedings*, pages 227–233. IEEE, 2014.

[27] Britta Hummel, Werner Thiemann, and Irina Lulcheva. Scene understanding of urban road intersections with description logic. In *Dagstuhl Seminar Proceedings*. Schloss Dagstuhl-Leibniz-Zentrum fr Informatik, 2008.

[28] Lihua Zhao, Ryutaro Ichise, Tatsuya Yoshikawa, Takeshi Naito, Toshiaki Kakinami, and Yutaka Sasaki. Ontology-based decision making on uncontrolled intersections and narrow roads. In *2015 IEEE intelligent vehicles symposium (IV)*, pages 83–88. IEEE, 2015.

[29] Jorge Godoy, Antonio Artuñedo, and Jorge Villagra. Self-generated osm-based driving corridors. *IEEE Access*, 7:20113–20125, 2019.

[30] Mordechai Haklay and Patrick Weber. Openstreetmap: User-generated street maps. *IEEE Pervasive Computing*, 7(4):12–18, 2008.

[31] Guillaume Bresson, Zayed Alsayed, Li Yu, and Sébastien Glaser. Simultaneous localization and mapping: A survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[32] Yao Fu, Hamid Moballegh, Raúl Rojas, and Longxu Jin. Reality sim: A realistic environment for robot simulation platform of humanoid robot. In *The 5th International Conference on Automation, Robotics and Applications*, pages 283–287. IEEE, 2011.

[33] Christian Erbsmehl. Simulation of real crashes as a method for estimating the potential benefits of advanced safety technologies. In *21st International Technical Conference on the Enhanced Safety of Vehicles (ESV) National Highway Traffic Safety AdministrationBundesministerium fuer Verkehr, Bau und Stadtentwicklung-BMVBS-(Federal Ministry of Transport, Building and Urban Affairs) Bundesanstalt fuer Strassenwesen (BASt)*, number 09-0162, 2009.

[34] Reinaldo Squillante Jr, Diolino J Santos Fo, Newton Maruyama, Fabrício Junqueira, Lucas A Moscato, Francisco Y Nakamoto, Paulo E Miyagi, and Jun Okamoto Jr. Modeling accident scenarios from databases with missing data: A probabilistic approach for safety-related systems design. *Safety science*, 104:119–134, 2018.

[35] Jinwei Zhou and Luigi del Re. Safety verification of adas by collision-free boundary searching of a parameterized catalog. In *2018 Annual American Control Conference (ACC)*, pages 4790–4795. IEEE, 2018.

[36] Aitor Arrieta, Shuai Wang, Urtzi Markiegi, Goiuria Sagardui, and Leire Etxeberria. Employing multi-objective search to enhance reactive test case generation and prioritization for testing industrial cyber-physical systems. *IEEE Transactions on Industrial Informatics*, 14(3):1055–1066, 2018.

[37] Marc René Zofka, Sebastian Klemm, Florian Kuhnt, Thomas Schamm, and J Marius Zöllner. Testing and validating high level components for automated driving: simulation framework for traffic scenarios. In *2016 IEEE Intelligent Vehicles Symposium (IV)*, pages 144–150. IEEE, 2016.

[38] Alessio Gambi, Tri Huynh, and Gordon Fraser. Automatically reconstructing car crashes from police reports for testing self-driving cars. In *Proceedings of the 41st International Conference on Software Engineering: Companion Proceedings*, pages 290–291. IEEE Press, 2019.

[39] Aric A. Hagberg, Daniel A. Schult, and Pieter J. Swart. Exploring network structure, dynamics, and function using networkx. In Gaël Varoquaux, Travis Vaught, and Jarrod Millman, editors, *Proceedings of the 7th Python in Science Conference*, pages 11 – 15, Pasadena, CA USA, 2008.