

STORYNET - A 5W1H-BASED KNOWLEDGE GRAPH TO CONNECT STORIES

A Thesis  
IN  
Computer Science

Presented to the Faculty of the University  
of Missouri–Kansas City in partial fulfillment of  
the requirements for the degree

MASTER OF SCIENCE

by  
SRICHAKRADHAR REDDY NAGIREDDY

University of Missouri

Kansas City, Missouri  
2021

© 2021

SRICHAKRADHAR REDDY NAGIREDDY

ALL RIGHTS RESERVED

# STORYNET - A 5W1H-BASED KNOWLEDGE GRAPH TO CONNECT STORIES

Srichakradhar Reddy Nagireddy, Candidate for the Master of Science Degree

University of Missouri–Kansas City, 2021

## ABSTRACT

Stories are a powerful medium through which the human community has exchanged information since the dawn of the information age. They have taken multiple forms like articles, movies, books, plays, short films, magazines, mythologies, etc. With the ever-growing complexity of information representation, exchange, and interaction, it became highly important to find ways that convey the stories more effectively. With a world that is diverging more and more, it is harder to draw parallels and connect the information from all around the globe. Even though there have been efforts to consolidate the information on a large scale like Wikipedia, Wiki Data, etc, they are devoid of any real-time happenings. With the recent advances in Natural Language Processing (NLP), we propose a framework to connect these stories together making it easier to find the links between them thereby helping us understand and explore the links between the stories and possibilities that revolve around them.

Our framework is based on the 5W + 1H (What, Who, Where, When, Why, and

How) format that represents stories in a format that is both easily understandable by humans and accurately generated by the deep learning models. We have used 311 calls and cyber security datasets as case studies for which a few NLP techniques like classification, Topic Modelling, Question Answering, and Question Generation were used along with the 5W1H framework to segregate the stories into clusters. This is a generic framework and can be used to apply to any field. We have evaluated two approaches for generating results - training-based and rule-based. For the rule-based approach, we used Stanford NLP parsers to identify patterns for the 5W + 1H terms, and for the training based approach, BERT embeddings were used and both were compared using an ensemble score (average of CoLA, SST-2, MRPC, QQP, STS-B, MNLI, QNLI, and RTE) along with BLEU and ROUGE scores. A few approaches are studied for training-based analysis - using BERT, Roberta, XLNet, ALBERT, ELECTRA, and AllenNLP Transformer QA with the datasets - CVE, NVD, SQuAD v1.1, and SQuAD v2.0, and compared them with custom annotations for identifying 5W + 1H. We've presented the performance and accuracy of both approaches in the results section. Our method gave a boost in the score from 30% (baseline) to 91% when trained on the 5W+1H annotations.

APPROVAL PAGE

The faculty listed below, appointed by the Dean of the School of Computing and Engineering, have examined a thesis titled “StoryNet - A 5W1H-based Knowledge Graph to Connect Stories,” presented by Srichakradhar Reddy Nagireddy, candidate for the Master of Science degree, and hereby certify that in their opinion it is worthy of acceptance.

Supervisory Committee

**Yugyung Lee, Ph.D.**, Committee Chair

Department of Computer Science Electrical Engineering,  
School of Computing and Engineering

**Ye Wang, Ph.D.**

Department of Communication Studies,  
College of Arts and Sciences

**Brent Never, Ph.D.**

Public Affairs,  
Henry W. Bloch School of Management

## CONTENTS

ABSTRACT . . . . .	iii
ILLUSTRATIONS . . . . .	viii
TABLES . . . . .	xi
ACKNOWLEDGEMENTS . . . . .	xiii
Chapter	
1 Introduction . . . . .	1
1.1 Overview . . . . .	1
1.2 Objectives . . . . .	7
1.3 Motivation . . . . .	10
1.4 Challenges . . . . .	12
1.5 Problem Statement . . . . .	13
1.6 Proposed Model: StoryNet . . . . .	13
1.7 Summary of Contributions . . . . .	17
2 Related Work . . . . .	18
2.1 Overview . . . . .	18
2.2 5W1H Component Detection . . . . .	18
2.3 Topic Modeling . . . . .	24
2.4 Time Series Forecasting . . . . .	25
2.5 Question Answering . . . . .	27

3	The 5W1H Framework and the Models Included . . . . .	29
3.1	Model Design . . . . .	29
3.2	“What” Model: LDA+BERT Custom Topic Model . . . . .	37
3.3	Objective 1. 5W1H component detection . . . . .	49
3.4	“When” Model: Time Series Forecasting . . . . .	52
3.5	“Why” & “How” Model: ELECTRA BERT / ALBERT / RoBERTa / T5 . . . . .	57
3.6	Transformer-Based QA Systems . . . . .	59
3.7	Objective 2: Graph Generation . . . . .	66
4	StoryNet Application: Evaluation and Results . . . . .	70
4.1	Case Study 1: 311 Calls . . . . .	70
4.2	Case Study 2: OCEL.AI . . . . .	130
4.3	Case Study 3: CVE (Common Vulnerabilities and Exposures) . . . . .	146
4.4	Case Study 4: CASIE (CyberAttack Sensing and Information Extraction) . . . . .	151
4.5	Case Study Comparison . . . . .	160
5	Conclusion and Future Work . . . . .	163
5.1	Conclusion . . . . .	163
5.2	Future Work . . . . .	164
	REFERENCE LIST . . . . .	166
	VITA . . . . .	183

## ILLUSTRATIONS

Figure		Page
1	Growing Complexity of Collective Behavior of Human Organizations [7]	3
2	Example: 5W1H Components . . . . .	8
3	Example: 5W1H Components in a Story . . . . .	9
4	Problem Statement . . . . .	14
5	Rule Based 5W1H Extraction . . . . .	19
6	Inverted Pyramid Technique [68] . . . . .	23
7	Architecture of StoryNet 5W1H Model . . . . .	30
8	Annotated Text . . . . .	31
9	Transformer Model . . . . .	32
10	BERT - Fine Tuning . . . . .	37
11	5W1H-StoryNet Model . . . . .	38
12	Architecture of LDA+BERT Clustering . . . . .	39
13	Topic Coherence . . . . .	40
14	Dominant Topic for 311 Record . . . . .	41
15	Topic Clusters - UMAP . . . . .	47
16	Topic Distribution in Documents . . . . .	48
17	Auto Annotation . . . . .	51
18	Time Series Analysis . . . . .	54



19	311 Service Category Wise Time Prediction . . . . .	55
20	311 Call Time Series Validation . . . . .	56
21	ELECTRA Architecture . . . . .	64
22	311 StoryNet . . . . .	65
23	Performance Comparison - Models . . . . .	66
24	Graph Generation . . . . .	67
25	Graph Connection Algorithm . . . . .	68
26	311 Data . . . . .	72
27	Visualization for 311 Call Map with Spatial Partition . . . . .	73
28	Map Visualization . . . . .	74
29	311 Call Volume Trends for Years from 2007 to 2020 . . . . .	77
30	Yearly Composition of 311 Call Types . . . . .	78
31	311 Calls - Round the Clock. . . . .	78
32	Departmental Call Volume Trend . . . . .	79
33	Categorical Call Volume Trend . . . . .	80
34	Architecture of Who & Where BERT Auto Annotation Model . . . . .	85
35	StoryNet Illustration . . . . .	96
36	311 StoryNet . . . . .	97
37	“Where” Class-Wide Classification Accuracy . . . . .	105
38	“Who” Class-Wide Classification Accuracy . . . . .	106
39	Confusion Matrix: (a) Where (b) Who . . . . .	107
40	311 Service Category Wise Time Prediction . . . . .	108

41	311 Call Time Series Validation . . . . .	109
42	Coherence for Optimal Topic Modeling . . . . .	112
43	Keyword Cloud . . . . .	118
44	311 - Topic Modelling . . . . .	122
45	StoryNet - Advantage . . . . .	131
46	OCEL.AI StoryNet Design . . . . .	137
47	OCEL.AI StoryNet . . . . .	144
48	CVE - Data . . . . .	147
49	CVE - Topic Cluster . . . . .	148
50	CVE - LDA Topics . . . . .	149
51	CVE - Word Cloud . . . . .	150
52	CVE - StoryNet . . . . .	151
53	CASIE - Question Generation . . . . .	154
54	CASIE - Data . . . . .	155
55	CASIE - WordCloud . . . . .	158
56	CASIE - StoryNet . . . . .	159
57	Performance Comparison - Use Cases . . . . .	161
58	Application Architecture . . . . .	162

## TABLES

Tables		Page
1	5W1H Questions . . . . .	4
2	5W1H Answers . . . . .	7
3	311 Time Series Predictive Model Accuracy . . . . .	58
4	KCMO 311 Service Request Dataset . . . . .	72
5	5W1H Components Identification - 311 . . . . .	90
6	Performance Comparison - GLUE Test . . . . .	101
7	Score Breakdown . . . . .	102
8	311 Who and Where Prediction Accuracy . . . . .	106
9	311 Time Series Predictive Model Accuracy . . . . .	110
10	Top 12 Topics and Topic Terms . . . . .	113
11	Silhouette Scores (SS) of Topic Models . . . . .	115
12	AI-Community-311-Question-Answers . . . . .	124
13	Technologies used in StoryNet modeling . . . . .	126
14	Topics and Dominant 311 Service Categories . . . . .	127
15	311 Service Categories Examples . . . . .	128
16	Topics and Minor 311 Service Categories . . . . .	129
17	ALBERT's Question Answering in 311 Services . . . . .	139
18	CVE Results . . . . .	152

19	CASIE Results . . . . .	159
20	Technologies Used in Apps . . . . .	161

## ACKNOWLEDGEMENTS

I want to thank my research advisor Dr. Yugyung Lee for all the support and guidance offered over the last two years. Dr. Lee's guidance helped me improve my knowledge in certain aspects of research and contribute to some of the challenging real-world problems. Deep learning is a new and growing field which have unlimited potential to solve many of the problems specific to automation. It is only with amusement and thanks to Dr. Lee to bring the new world technologies as courses with intuitive course learning helping students to be ready to solve some of the complex challenges. I'm consistently amazed by Dr. Lee's research drive and the kind of attention given to details despite handling several research problems at once.

I would like to take this opportunity to thank the individuals of my thesis committee, Dr. Yugyung Lee, Dr. Brent Never, and Dr. Ye Wang. I am fortunate to have worked on community research for Kansas City. It helped me to see a new perspective and conduct the research further to solve the problem. Thanks to the University of Missouri-Kansas City for providing the kind of support it did during tough times we have seen through 2020-21. The opportunities provided by UMKC helped me master new skills, and it's one of the finest decisions I made to pursue my education with UMKC.

I would like to acknowledge the partial support of the NSF, USA Grant No. 1747751, 1935076, and 1951971. I would like to thank my family for all the support and guidance I have received during the tough time. I extend great thanks to all friends and mentors who were part of my journey in these two years.

## CHAPTER 1

### INTRODUCTION

#### 1.1 Overview

In this thesis, we present a framework based on 5W1H (Who, What, When, Where, Why and How), to connect stories from multiple times, domains, and sources together on common features among those stories using techniques in Natural Language Processing like Question Answering, Topic Modelling, and Text embeddings.

In [7], Y. Bar-Yam has illustrated the evolution of information exchange and the interactions among humans from the early days of the hunter-gatherer era to the current state of civilization as shown in Figure 1. In the early days, we didn't have as many connections because we didn't need to exchange as much data. There were not as many sources or possibilities for long-distance communication. But as we grew in population and started making more connections, we formed new networks and exchanged information locally. As the networks started growing in number and size, they started coming together and became bigger networks with intertwined causes and effects on a bigger scale; as we can see today, something that happens in one part of the world can affect many other parts of the world.

All the networks that are connected may not be grouped together spatially or temporally. The study also shows the increase in diversity and variety of the information exchanged through time. More domains and specializations have developed giving rise

to diversity in the information grouping and connecting, making it more complex to represent a phenomenon or an idea within just one group or network. For example, initially, health care and travel may not be entirely different in a domain, but now there are more and more lines that are being drawn between such different domains. There is a relation or connection between almost all the stories that are shared some way or the other. Even though they are connected on a similar thread, they can not be often identified together, i.e., in the same source, or the same geography or the same time.

This means that some story that happens today in some part of the world may have an effect on another story in another part of the world or another point in time. This may not be evident right away until we look at them at once and connect the dots. For example, this would help a CEO of a multi-national, multi-disciplined company who needs to be knowledgeable across teams with cross discipline insights in the company. Another example would be a medical researcher who wants to have the information available at hand that he/she can navigate through to predict the next pandemic.

This thesis proposes a framework with a connected graph that let's us navigate through the stories fluidly and also answer questions based on the data. It features a Question Answering engine that makes use of a re-trained BERT model to extract the answers for the 5W1H components. These components are connected together using a semantic matching algorithm on the graph, which is explained in the later sections.

5W1H is used as the representation for the framework that is being proposed. It is one of the most universally used tools for information gathering, analysis, organization and presentation. This method is used across a range of professions, from process analysts

# Historical Progression

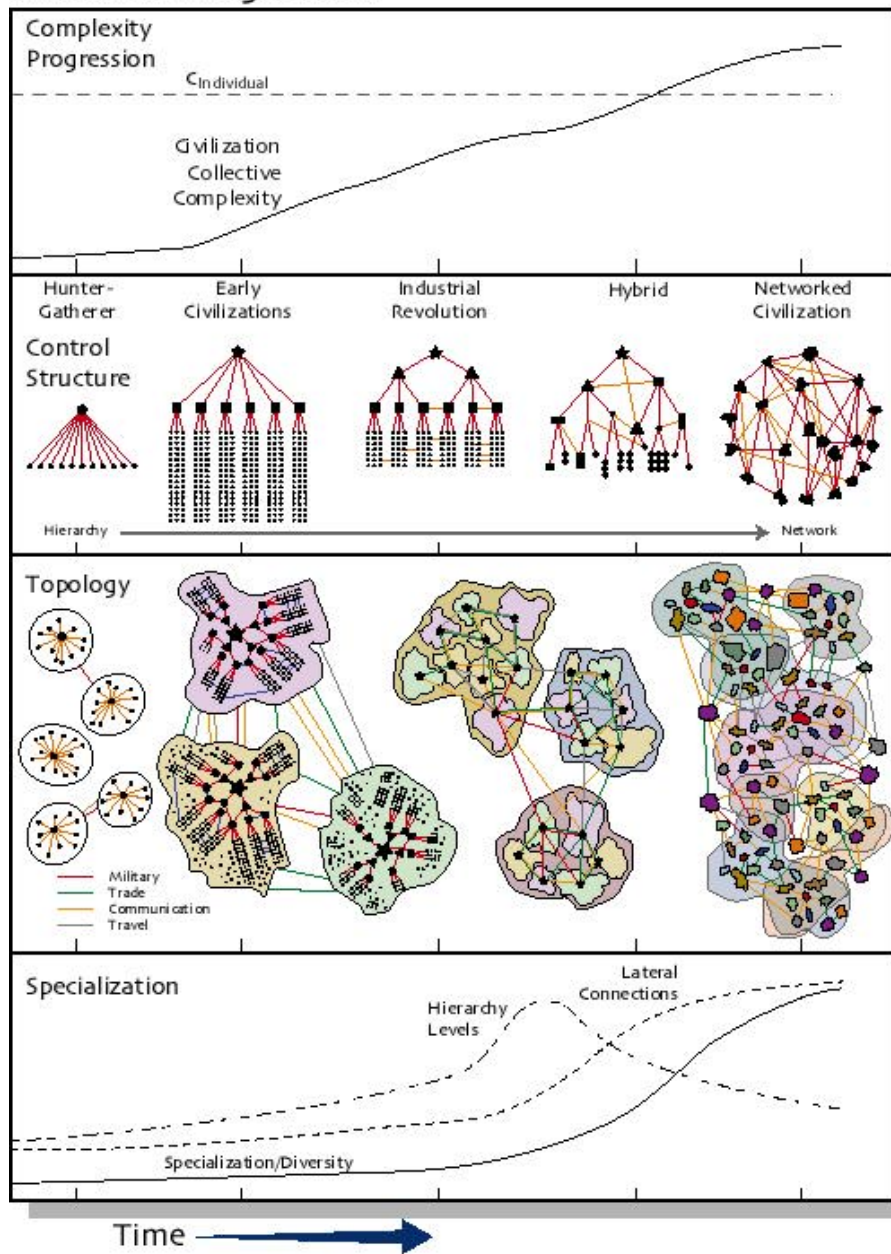


Figure 1: Growing Complexity of Collective Behavior of Human Organizations [7]



<b>Who</b>	Who was involved?
<b>What</b>	What happened?
<b>When</b>	When did it happen?
<b>Where</b>	Where did it happen?
<b>Why</b>	Why did it happen?
<b>How</b>	How did it happen?

Table 1: 5W1H Questions

to quality engineers to journalists, to understand and explain virtually any problem or issue. The same method can be used to organize the writing of reports, articles, white papers, and even whole books.

#### 1.1.1 The Basic Approach

This approach seeks to answer six basic questions in gathering information about nearly any subject: Who, What, When, Where, Why, and How. The concept of 5W1H was first introduced by Griffin Et al. [34], and is widely used in journalism. In journalism, a news article or a story is considered to be complete and correct only when the 5W1H are present. The 5W1H provide the facts about a news article or a story being written such as:

In journalism, news story writing requires that these questions are answered to take a basic form. Applying the 5W1H framework to other types of writing or investigation takes some interpretation. The order in which the answers to the questions is presented may vary, but the “what” is usually addressed first.

**What:** In journalism, the “what” identifies an event and is often stated in the “lead (or lede),” the opening sentence or paragraph of a news article, summarizing the most

important aspects of the story. The “what” is the primary subject, the reason the information is being gathered and presented. Apart from journalism, it may be stated in a title and in a purpose statement. The “what” may need to be defined, a process that may comprise the remainder of a document.

*Example: What, specifically,...?*

**Who:** A news story identifies who an event involves. The “who” may be part of the lede, and could be the reason the story is news worthy. In other contexts, the “who” identifies the persons or groups the “what” concerns. It might describe the audience of a document, or those who are affected by a policy, process or procedure.

*Example: Who benefits?*

**When:** A key part of a news story is describing when an event happened. Answering the “when” indicates any time sensitivity related to the “what.” It may be part of an instruction regarding the proper point at which a action should be taken. Sometimes it may be part of an “If...then” scenario of conditional action.

*Example: When will it start/end?*

**Where:** A news story reports the location at which an event took place. The “where” describes a geographical or physical location of importance to the “what.” At times, the where may be less important than other factors.

*Example: Where are you?*

**Why:** The “why” is usually the most neglected of the questions in the framework. News

stories often lack information from authoritative sources to explain the “why.” In other contexts, the “why” may be considered irrelevant, particularly when describing a policy or procedure decreed by an organizational authority. Efforts to ascertain and explain the “why” may help those affected be more accepting of any change the “what” requires.

*Example: Why does that happen?*

**How:** For journalists, determining how an event took place may be nearly as challenging as explaining the “why,” although more effort is usually put to satisfying the question. When describing policies, processes or procedures, the how may be the most important part of the effort. A considerable appetite for understanding how to do something can be found across audiences. Sometimes effort focuses on the “what” when more work should be devoted to explaining the “how.”

*Example: How much?*

The 5W1H framework can be applied to any topic at any level of granularity to gather, analyze and present information from the simplest to the most complex. Attributed to a Rudyard Kipling poem, 5W1H is the place to start and may be enough to take you to the finish. For 5W1H annotation, we adopted a Question and Answer (QA) based approach similar to [37] and [105] to extract the answers to the 5W1H questions. Let us consider the event of the recent US elections.

**Example:** *Joe Biden won the elections in 2020 with 53% majority for US administration.*

<b>Who</b>	Joe Biden
<b>What</b>	won the elections
<b>When</b>	2020
<b>Where</b>	US
<b>Why</b>	administration
<b>How</b>	53% majority

Table 2: 5W1H Answers

In the above example, the answers to 5W1H questions are shown in Table 1.1.1 and are represented as 5W1H components of the story in a graph in Figure 2.

This segmentation step of our framework requires the identification and classification of 5W1H components. We, therefore, used [34] as the framework for our attention based deep neural network system for 5W1H component identification and classification task. There is one more example illustrated in Figure 3 with a few missing components. The solution to this missing components will be addressed in Chapter 3’s Model Section.

## 1.2 Objectives

In this thesis, we have addressed three objectives.

Objective 1: 5W1H Component Detection

Objective 2: Building StoryNet

Objective 3: StoryNet Application

**Objective 1. 5W1H component detection:** The first objective is to detect the 5W1H components by using automatic annotation. A BERT based deep learning model

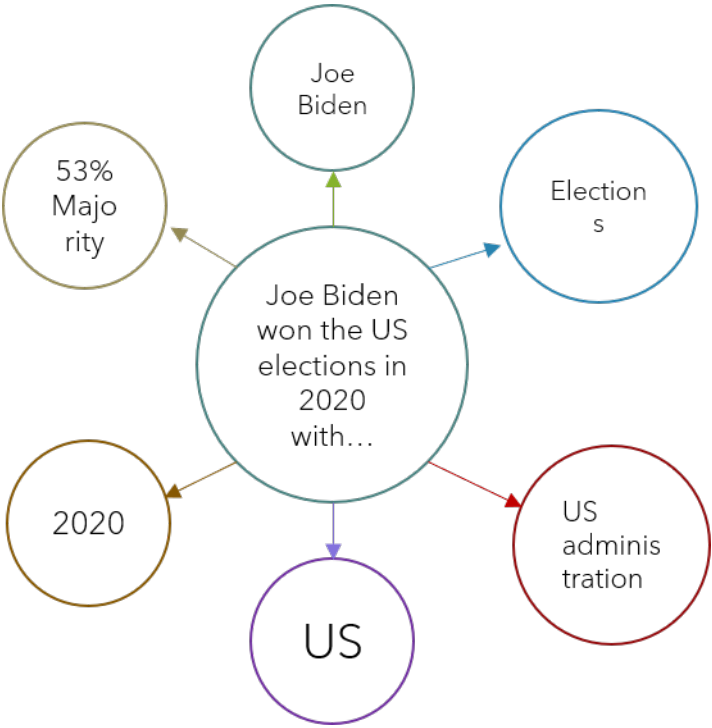


Figure 2: Example: 5W1H Components

## The Basic News Lead

A 16-year-old was arrested in the robbery and beating death of an 88-year-old veteran from Spokane, Wash., who survived being wounded in World War II.

- **Who:** A 16-year-old
- **What:** Was arrested
- **When:** ??
- **Where:** Spokane, Wash.
- **Why:** robbing and beating an 88-year-old veteran to death
- **How:** ??

Figure 3: Example: 5W1H Components in a Story

is used to train for 5W1H detection using manual and automatic annotations. Automatic annotations are the answers to the 5W and 1H questions provided by the deep learning models based on BERT. The classification models provide the answers to “who” and “where” questions, modeling model provides the answers to “what” question and the time series model provides the answer to the “when” question. The answers to the “how” and “why” questions are provided by the question answering model.

These answers are then used to mark the annotations in the original paragraph or text and then are used to train the 5W1H model. This reduces the amount of time and human effort that is needed for annotation which is considered as one of the

biggest bottlenecks in training a deep learning model. Automatic annotations help us to reduce the amount of time it takes for the annotation part and allows us to get the results faster while not compromising on the accuracy.

**Objective 2. Building StoryNet** The extracted find 5W1H components are connected together in a graph using a graph matching algorithm which makes use of the semantic similarity score. Neo4j is used to store the 5W1H component and the graph algorithm is run using the programming language Cypher which is used as the primary language to interact with the Neo4j graph database. This is a knowledge graph that lets the user to navigate easily between the stories showing their common components and helps to identify and draw parallels between the stories using the connected 5W1H components.

**Objective 3. StoryNet Application** The final objective is the application of storing it too if you use cases like 311 calls, Cyber security news articles, OCEL.AI articles. We have evaluated the performance of the model based upon an ensemble of multiple scores used in the natural language processing domain. The advantage of this process is the ease of adaptation to a domain using automatic annotation and it has outperformed the other models which make use of genetic domain knowledge.

### 1.3 Motivation

**Scale of information:** There has been an incredible growth in the number of stories that have been published lately. Research from NSF (National Science Foundation) shows that the worldwide S&E publication output volume continues to grow on an

average at nearly 4% an year. Just from 2008 to 2018 the output grew from 1.8 million to 2.6 million articles [109].

**Time to analyze:** According to [28], it takes weeks if not months, for content analysis on just on research paper. So analyzing a ton of research papers and connecting the dots between them and putting them together is a seemingly very time taking and strenuous task to be performed by any human being. Making use of the natural language processing techniques for this task would save a ton of time for analysis and also provide the information that is important in a time critical situation.

**Lack of whole picture:** More often than not, we have the information presented in multiple studies, that can be grouped together instead of a single resource. This is the reason why we have aggregations of research papers, news articles, and any other sources grouped together in certain places like papers with code (paperswith-code.com), connected papers (connectedpapers.com), etc. Connecting the stories together makes it easier to find and grow the stories that are similar to each other which have a common component among them.

**Scattered Sources:** All of the information related to a particular story needs to be gathered by someone from multiple sources and put together to understand it holistically. This task is currently being performed by agencies that have people who hunt for news and related articles and put them together as documentaries or news reports. There are 1761 commercial television stations in 2018, which grew from a mere 98 in 1950, 1279 daily newspapers and 1.7 billion websites only in the USA



according to Statista [94]. It would be a great help to automate this process and put the information together using natural language processing techniques.

**Comprehensibility:** Comprehending the information is one of the biggest researches that has triggered a lot of applications in the field of natural language processing. For example research in paraphrasing, summarization and question answering has helped to create new architectures and models with better performance in those tasks. Graph is one of many ways to comprehend a seemingly large amount of information.

## 1.4 Challenges

These are a few technical challenges that we came across for which we had to come up with solutions, some of which are simple choices that need to be made among all the available ones and others that needed to come up with an algorithm.

**Representation of a Story:** A story can be represented in many ways - using techniques like comprehension, summarization, question answering (frequently asked questions), etc. We chose to represent it using 5W1H framework.

**Annotations:** Manual annotations are the bottleneck for any deep-learning training based application. We improvised the time it takes, using automatic annotation with classification, topic modeling and time series analysis models.

**Connecting Stories:** We need a common thread to connect two stories. In our solution, 5W1H components serve the purpose of connecting the stories. We proposed a

novel graph connecting algorithm that makes use of a semantic similarity score.

**Graph Search:** Searching a graph that is huge, takes a lot of compute power and time.

We addressed the problem by making use of a dictionary and mapping it the nodes of the graph and using the dictionary to perform the search, which makes it much faster and less complicated.

## **1.5 Problem Statement**

The problem statement of this thesis is to add structure (5W1H) to unstructured data. The unstructured data can be any text preferably a news article, research paper, description of a cyber attack, 311 call / report, or just a piece of text. The proposed 5W1H framework represents the text or a story as a graph and also connects it to other stories based on the similarities between them.

The steps of annotation, component identification and connection are automated by making use of natural language processing techniques. The final output graph helps anyone to easily navigate and understand the unstructured data with a minimal effort. And overall idea of the process is illustrated in Figure 4.

## **1.6 Proposed Model: StoryNet**

### **1.6.1 Formal Definition of Knowledge Graphs**

Applying the domain-in-the-loop approach, we now define the concept of domain StoryNet (StoryNet). StoryNet is composed of real-world entities, their relations and



Figure 4: Problem Statement

domain knowledge. For example, in Cybersecurity, three main parties are involved: cybersecurity experts, cyber attackers, and victims. StoryNetwork will be designed for real-world problems and solutions for social bots attacks. Complex scenarios of social bots involve 1) human attackers conspiring with AI to inflict harms, 2) human cybersecurity agents working with AI to monitor and predict vulnerabilities, fend off attacks, and repair damages, and 3) the targeted human victims to take precautionary measures or to report suspicious/actual attack events.

Each story will be organized around these main characters: (a) **Who** are 1) the human cybersecurity agent (the learner) and AI (the machine), 2) the targeted victims, and 3) the attackers. (b) **What** includes 1) vulnerabilities of the targeted victims and the trained deep learning (DL) models, and 2) the best practices that guard against those vulnerabilities. This includes not only what the human cybersecurity agent should do, but also what the targeted victims should be doing to avoid harms.

The structure of the storytelling approach features a network of stories, each of which contains three prongs: 1) various vulnerabilities, 2) possible social bots attacks, and 3) best practices to deal with potential and actual threats. An example is phishing emails sent to university employees: The first prong is about those who may be targeted, including those who may be impersonated or those who may receive the phishing messages. The second prong is about the attackers' strategies and techniques to execute the attacks. As the cyberattacks get more personalized and targeted, we will use public information with the consent from the users, to simulate personalized and targeted attacks and create interactive storytelling.

The third prong is about the cybersecurity agent and AI, including hands-on examples of the best human-AI practices for cybersecurity. Each mini-story contains a complete three-prong narrative, accompanied by student learning outcomes and assessment tools. Mini stories are semantically connected with each other using Natural Language Processing, to allow dynamic interaction with the network of stories.

StoryNet also specifies connections among entities, and describes the relations between the nodes in StoryNet. In a domain-in-the-loop scenario, the communications and interactions that have been conducted by the team through news media or social networks (e.g., sharing news). The StoryNet network represents instances of stories and their relations to the domain. The 5W1H relationships will be discovered through the partnership with humans and machines. We will represent the discovered relations as a form of knowledge graphs using the well-known story frame “5W+1H.” The reason for the use of the story frame is to make it more intuitive and accessible for real-world applications, question answering or recommendation, and chatbots.

Thus, StoryNet is more than information network or transaction networks but this is core to understand the problems in the domain or analyze their impacts to the domain from the reasoning with the StoryNet network. More interestingly, this StoryNet networks will be built through the domain-in-the-loop approach. The StoryNet network can be built for different domains such as neighborhoods, business, education, healthcare. In this thesis, we will present the StoryNet in cybersecurity domain domain.

## 1.7 Summary of Contributions

In summary, there are three main contributions that are a part of this thesis are:

- Identifying and generating automatic annotations for the 5W1H components in the unstructured data using deep learning models.
- Segmentation and representation of the unstructured data as 5W1H component graph.
- Algorithm to index and connect the identified 5W1H components for all the stories in the unstructured dataset.
- Application of the StoryNet itself on a dataset and evaluate the result.

## CHAPTER 2

### RELATED WORK

#### 2.1 Overview

Analyzing unstructured text data came to light in the recent years due to the advancements in natural language processing techniques like attention mechanism, transformers and other related fields of study - like information extraction, topic modelling, question answering, language modelling and multi-modal representations. The authors [101] started a revolution in the domain. It gave birth to transformers, which became the norm for a plethora of architectures in topic modelling and question answering. Described below are the works that led to the development in text analytics which made StoryNet possible.

#### 2.2 5W1H Component Detection

Identifying the 5W1H components are critical to the creation of StoryNet as they are the building blocks for an application. The existing approaches for detecting 5W1H mostly solve the problems using a rule based approach as shown in Figure 5. Kunal Chakma et al., proposed a semantic role labelling (SRL) approach in [14], which makes use of several lexical resources available for SRL such as PropBank [72], FrameNet [88], VerbNet [5] to identify the predicate and manual question answering to identify candidates for 5W1H components. They have segmented sentence into different parts where, action

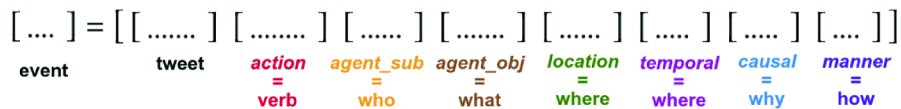


Figure 5: Rule Based 5W1H Extraction

corresponds to a verb, and the rest of answers depend on the parts of the grammar tree.

Shi et al. [91] discussed about using LSTM for extending BERT and training the custom model can be adapted to relation extraction and semantic role labeling without syntactic features and human-designed constraints. This idea is extended in this study to make use of a faster custom fully connected layer instead of LSTM which is slower. Long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture [58] used in the field of deep learning. Unlike standard feed-forward neural networks, LSTM has feedback connections. It can process not only single data points (such as images), but also entire sequences of data (such as speech or video).

Recently, significant advances have been made in Natural Language Processing and Deep Learning, such as ELMO [74], GPT family [13, 78, 79], BERT [24], RoBERTa [53], XLNet [115] and ELECTRA [21]. In particular, BERT has been extensively explored in conjunction with various NLP models to achieve state-of-the-art performance. The pretrained uncased BERT is introduced in Devlin et al. [24]. It generates representations from unlabelled text data by jointly conditioning on both left and right contexts (bi-directional) in all layers. It can be fine-tuned with just one additional output layer for various tasks.

Peinelt et al. [73]’s topic-informed BERT-based model (tBERT) combined topic representation of a sentence from LDA topic models with the sentence pair vector  $C$



$(C = BERT(S_1, S_2) \in R^d)$  of  $BERT_{BASE}$ .  $S_1$  is a sentence with length  $N$ , and  $S_2$  is another sentence with length  $M$ . Both are the uncased version of  $BERT_{BASE}$ , which does not differentiate between english and English. Employing  $BERT_{BASE}$ , Peinelt et al. [73]’s iBERT with LDA topics produced accurate and stable performance across a range of benchmark datasets of semantic similarity prediction. The pretrained uncased BERT is introduced in Devlin et al. [24]. It generates representations from unlabelled text data by jointly conditioning on both left and right contexts (bi-directional) in all layers. It can be fine-tuned with just one additional output layer for various tasks.

In contrast to Peinelt et al. [73]’s model-driven approach, Venkataram et al. [102]’s experimentation with LDA and BERT was largely data-driven. Answering the call from the White House, they aimed at exploiting the COVID-19 Open Research Dataset consisting of more than 29,000 machine-readable articles on COVID. Their TopiQAL was an “interpretable, unsupervised, generic and fused” ML and DL architecture for COVID-related question-answering.

They used LDA and BioBERT, and their unique contribution was the hierarchical inference that matches user query sentences with LDA topic distribution of abstracts with a probability threshold = 0.2, followed by the same process on paragraphs in body text. Two levels of topic model filtering supplied chosen topics to BERT extractive summarizer for Q&A. BioBERT was BERT adapted for the biomedical domain [49].

The existing studies showed that applying the BERT model can generally improve the performance of NLP tasks with appropriate adaptation and fine-tuning. The current study will experiment with topic models with BERT in the domain of 311 calls. Using the

BERT pre-trained model, many NLP applications have been developed through transfer learning from the pre-trained model to the target domain by fine-tuning. e.g., Transformer [100], which add new layers for solving specific NLP tasks. BERT is an effective way to build a new model by selecting suitable parameters. However, there is a lack of well-assessed best practices to achieve high accuracy on real-world data. This study aims to investigate NLP and Deep Learning technologies in the classification and time-based prediction tasks to bring the communities into the loop.

A recurrent neural network (RNN) [58] is a class of artificial neural networks where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feed-forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs. This makes them applicable to tasks such as unsegmented, connected handwriting recognition or speech recognition.

Twevent [51] proposed a segmentation-based event detection system from news articles. Tweets were segmented using a segment score of “stickiness” in their approach, and bursty segments were selected based on segment prior probability distribution, and user diversity. They finally clustered the news articles into events. SEDTWik [62] is an extension of Twevent [51] with more features. SEDTWik used hashtags, retweet count, user popularity, and follower count as the key features. They achieved better results by giving more weightage to hashtags. Dabiri and Heaslip [90] presented a deep learning-based system for traffic event detection from the Twitter stream. They used both Recurrent

Neural Networks (RNN) [90] and Convolutional Neural Networks (CNN) [112] architectures on top of word embeddings. TwitterNews+ [36] is a system for event detection from the Twitter stream in real-time which integrates inverted indexes and incremental clusters to detect major as well as minor events considered to be newsworthy. They utilized several parameters such as M (number of most recent news articles), tSi (expiry time for an event cluster), tsr (cosine similarity threshold) and fine-tuned them to achieve the best performance.

Event trigger-based classification uses deep learning models based upon feature engineering that selects event trigger words [46]. Recent studies also investigated methods of combining event trigger words with contextual words to enhance the performance [66]. However, the event trigger-based classification often uses supervised machine learning, which causes difficulty in performing classification on unseen classes [46]. Recognizing the limitation of even trigger-based classification, Ngo et al. [66] applied few-shot learning event detection.

Above two methods focus mainly on data in the form of Twitter tweets and are not generic enough to be applied on any kind of unstructured text data. In [68], Keith et al, used an inverted pyramid score to evaluate the 5W1H sections of a text as illustrated in Figure 6. But they focused on the data mainly containing a very specific structure with the headline followed by the body with the content in an inverted pyramid structure. So it cannot be applied on data in any format which truly does not solve the problem of handling unstructured data.

These existing approaches of Event Determination are not immediately applicable

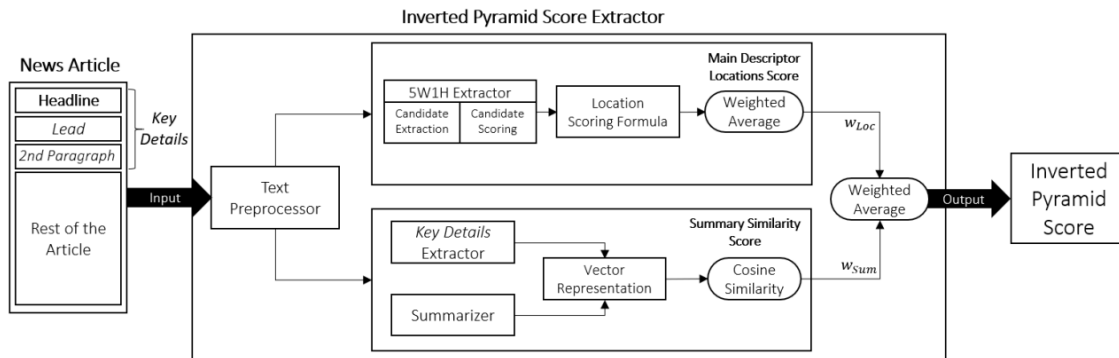


Figure 6: Inverted Pyramid Technique [68]

to our case. Mound and reconnect to be applied on data, and also they could not report accuracy or any other metrics because their rule-based instead of training-based. It is necessary to combine the supervised classification approach with the unsupervised automatic annotation, topic modeling and question answering approaches.

Unlike Twevent [51] and SEDTWik [62] where a segment of a tweet is an n-gram with no semantic structure, our 5W1H segmentation approach provides a highly semantically rich structure which helps in better clustering of the events. Though we modelled the events similar to 5WTAG [118], the two approaches are different in terms of their use. The primary emphasis of 5WTAG is to construct a candidate hashtag from the 5W segments whereas, our 5W1H based segmentation is oriented towards establishing the relationship between the predicates and the corresponding arguments and using this relationship for clustering the events.

### 2.3 Topic Modeling

We have proposed a topic prediction model by building the contextual topic approach based on the integration of LDA [10] and BERT [24]. This model aims to identify the topic that the model would develop and plot a corresponding word cloud for each of the  $K$  unique categories where the number of issues  $K$  was determined by the optimal topic model based on KL-divergence. The dominant topic for each description is identified by matching the probability of each topic with the description. These plots can be mapped to the 311 categories in our domain.

Latent Dirichlet allocation (LDA) [10] is a generative probabilistic model based on the three primary structures, including word, topic, and document. A distribution over topics is determined with the documents while a distribution over words with each topic. Given an input corpus  $D$  with  $V$  unique words and  $M$  documents, each document containing a sequence of words  $d w_1, w_2, \dots, w_{N_d}$ . Given an appropriate topic number  $K$ , the generative process for a document  $d$  is as following: Sample a  $K$ -vector  $\delta_d$  from the Dirichlet distribution  $p(\delta|\alpha)$ , where  $\delta_d$  is the topic mixture proportion of document  $d$ . For  $i = 1 \dots N_d$ , sample word  $w_i$  in the document  $d$  from the document-special multinomial distribution  $p(w_n|\delta_d, \beta)$ , where  $\alpha$  is a  $k$ -vector of Dirichlet parameters, and the Dirichlet distribution  $p(\delta|\alpha)$   $\beta$  is a  $K \times V$  matrix of word probabilities, where  $\beta_{-ij} = p(w_{-j} = 1 | z_{-i} = 1), i = 0, 1, \dots, K; j = 0, 1, \dots, V$ .

In LDA, the topic proportions are randomly drawn from a Dirichlet distribution, which implies the independence between topics. However, these correlations are widespread in real-world data. Interestingly, the association or correlation of topics can

be modeled with LDA. The topic “311 services” is often correlated with “crimes” while unlikely co-occurs with “business.” There would be an inconsistency between the assumption and input documents so that the predefined parameter  $K$  may not be able to reflect the real topics of the domain. To overcome the limitation, the NLP embeddings, such as BERT, could be significantly contributed to determining relevant or non-related topics in the 311 call domain.

## 2.4 Time Series Forecasting

The traditional univariate forecasting techniques predict future values of time series based on its historical values [39]. However, amid the criticism on the black-box nature of the artificial neural network, recent successes of recurrent neural network (RNN) models have shown great potential [39]. For example, Long short-term memory (LSTM), which is an artificial recurrent neural network (RNN), was trained and tested on datasets of COVID transmission in Canada and Italy to predict future outbreaks [20].

Time series forecasting is focused on modeling the predictors of future values of time series given their past. As in many cases, the relationship between past and future observations is not deterministic; this amounts to expressing the conditional probability distribution as a function of the past observations:  $p(X_{t+d}|X_t, X_{t-1}, \dots) = f(X_t, X_{t-1}, \dots)$ . (1) This forecasting problem has been approached almost independently by econometrics and machine learning communities.

Some studies found that deep learning models like LSTM produced more accurate and clear patterns and predictions than mathematical and statistical models [20]. They can

learn to identify non-linear patterns and explore latent relationships without any prior [93]. Smyl [93] mixed an exponential smoothing (ES) model with LSTM into one common framework that combines the strength of statistic models, and neural networks [93]. This hybrid forecasting method used exponential smoothing for deseasonalizing and normalizing the series and LSTM for extrapolating the series [93]. Livieris et al. [54] created a CNN-LSTM model to predict the price of gold. This approach uses CNN to preprocess data and screen out noises, and an LSTM layer is stacked on top of it to perform forecasting. Their first CNN-LSTM model without a fully connected layer performs well on regression tasks, like predicting prices. Their second CNN-LSTM model with a fully connected layer performs well on classification tasks like predicting the gold movement. Niu et al. [67] combined two-stage feature selection, convolutional LSTM, GRU, and an error correction model to predict the financial market.

Sirignano et al. [92] proposed a time series price prediction with a 4-layer perceptron model for price changes in Limit Order Books. Neil et al. [65] proposed an LSTM architecture for asynchronous series detection by tackling learning dependencies of various frequencies in the time series. Borovykh et al. [11] proposed a predictive model with convolutional neural networks for conditional time-series as these related studies show that real-world data often require a different combination of techniques. Assessing models on real-world data is needed to find out the best practices of performing time-series forecasting.

Borovykh et al. [11] proposed a predictive model with convolutional neural networks for conditional time-series, a WaveNet architecture with short univariate and bivariate time-series. This is a more recent WaveNet architecture [71] to several short univariate and bivariate time-series (including financial ones). Despite the claim of applying deep learning, Heaton et al. (2016) use autoencoders with a single hidden layer to compress multivariate financial data. Neil et al. (2016) present the augmentation of LSTM architecture suitable for asynchronous series, which stimulates learning dependencies of different frequencies through the time gate. In this thesis, we investigate the capabilities of several architectures (CNN, Residual Network, multi-layer LSTM, and Phased LSTM) on AR-like artificial asynchronous and noisy time series, household electricity consumption dataset, and on real financial data from the credit default swap market where some inefficiencies may exist, i.e., time series may not be totally random.

In this thesis, we will model the time series forecasting by expending the prophet [98], which is used in practical forecasting of business time series, for predicting response times for a specific 311 service by analyzing the time series of 311 service progress. The relationship between past and future observations of 311 service response times is modeled based on the conditional probability distribution.

## **2.5 Question Answering**

Question Answering systems in information retrieval are tasks that automatically answer the questions asked by humans in natural language using either a pre-structured database or a collection of natural language documents (Chali et al. [15], Dwivedi and



Singh [26], Ansari et al. [3], Lende and Raghuwanshi [50]). In other words, QA systems enable asking questions and retrieving answers using natural language queries (Abdi et al. [1]). Yu [116] considered QA systems an advanced form of information retrieval. The demand for this kind of system increases on a daily basis since it delivers short, precise and question-specific answers [76]. With the efforts from academic research, the QA subject has attracts growing interest around the world ( [103], [107]) and the main evidence of this is the IBM Watson [30].

This Systematic Literature Review (SLR) was based on guidelines provides by Okoli and Schabram [70], Keele [44]. The review tasks are based on their eight steps, and here we will describe: Purpose of the Literature Review, Searching the Literature, Practical Screen, Quality Appraisal and Data Extraction.

A exponential growth in written digital information led us to the need for increasingly sophisticated search tools (Bhoir and Potey [9], Pinto et al. [75]). The amount of unstructured data is increasing and it has been collected and stored at unprecedented rates (Chali et al. [15], Bakshi [6], Malik et al. [55]). The challenge is to create ways to consume this data, extract information and knowledge having an interesting experience in the process. In this context the Question Answering systems emerge, providing a natural language interaction between humans and computers to answer as many questions as possible and enabling the retrieval of these answers from unstructured data sets.

## CHAPTER 3

### THE 5W1H FRAMEWORK AND THE MODELS INCLUDED

#### 3.1 Model Design

In case of the StoryNet’s 5W1H model, the raw unstructured text is first annotated using automatic annotation as shown in Figure 7. Automatic annotation is achieved by the use of a few other models (classification, time series, topic modelling and question answering) that provide the answers for 5W1H questions. After identifying the answers, they are marked in the paragraph using an annotation tool or script. Automatic annotation is done to reduce the amount of time it takes to create the model thereby helping us to get faster results.

The annotated text is then pre-processed by passing through the pre-trained BERT [89] model which is a transformer-based [56] text embedding technique that makes use of attention [100]. This step generates text and meetings which are efficient to be used to train the custom deep learning model along with the annotation embeddings. An example of the annotated text is illustrated in Figure 8.

Attention techniques aim at having different embeddings to understand the sentences and their contexts within the sentences. This can be demonstrated by a simple example using an input sentence as follows: “The animal didn’t cross the street because it was too tired.” Instead of focusing on the whole input in the sentence, the attention mechanism abstracts the input embeddings into vectors as queries, keys, and values. The

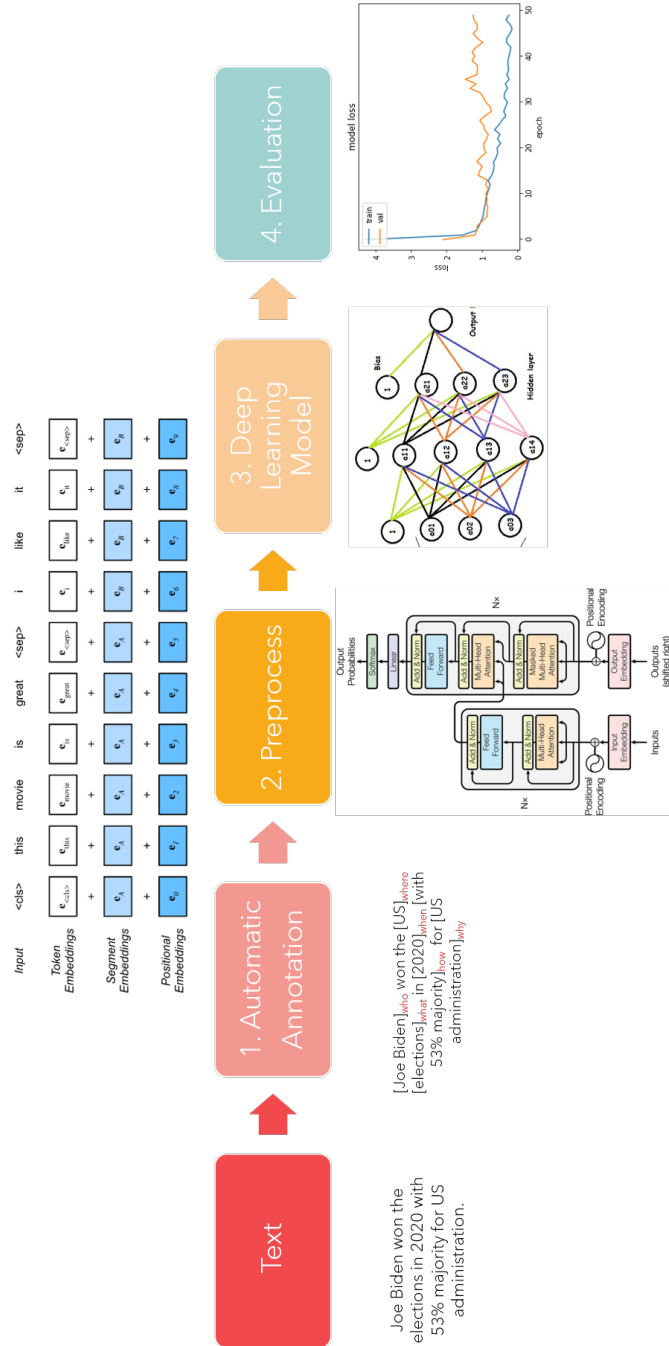


Figure 7: Architecture of StoryNet 5W1H Model

## Hawaii group wants to defend licenses for foreign fishermen

HONOLULU (AP) — A group representing Hawaii commercial fishermen has filed a court motion to defend the state's practice of giving fishing licenses to foreign workers.

The Hawaii Longline Association sought court permission last week to argue its side after a Maui resident asked a judge to declare that only those lawfully admitted to the United States should receive commercial fishing licenses.

Resident Malama Chun went to court after an Associated Press investigation found hundreds of foreign workers in the Hawaii fleet were confined to boats and some were living in subpar conditions.

[...] Who What When Where Why How

Figure 8: Annotated Text

vectors could be understood as search text, content text, and query text in the traditional searching mechanism.

- A few kinds of popular embeddings are described below:

**Input embedding:** This embedding aims to represent each word with a unique token and represent that into the model for attention technique to perform context parsing.

**Segment embedding:** This embedding aims to provide information about the words belonging to each sentence. So traditionally, two sentences are provided as input, where the segment embeddings for all the tokens in the first sentence are represented as one. The next sentence is represented as 2nd segment.

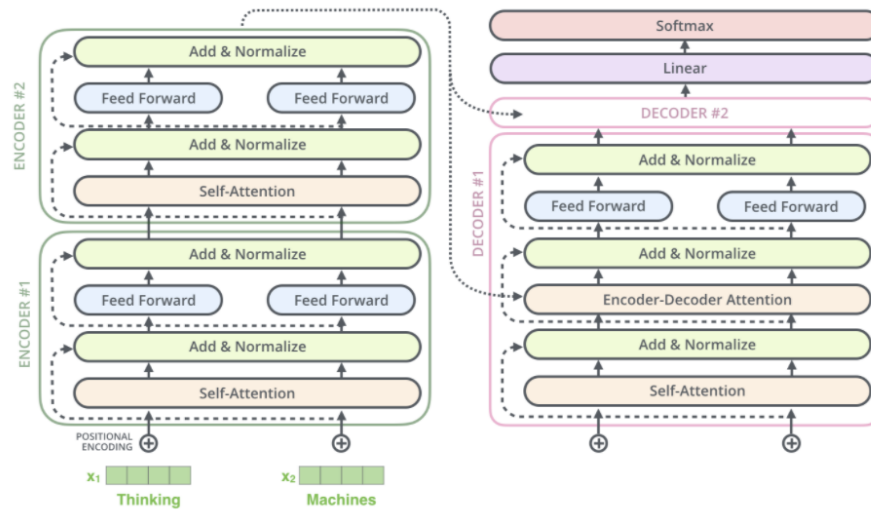


Figure 9: Transformer Model

**Positional embedding:** Positional embedding helps to understand the position of the tokens within the segments. So the positions would be tokenized with values starting 1 through the number of tokens in each sentence. So the positional embedding restarts from value 1 when starting a new sentence.

In the attention model, every input is evaluated into three vectors queries, keys, and values. The vectors represented as K, Q, and V have learned weights. The model built with an attention mechanism is called a transformer [56]. A transformer is represented as show in Figure 9.

### 3.1.1 “Who” and “Where” Model: BERT-based 5W1H Classification

There are two four different tasks where are carried out for each record of the data set to come up with the answers to each of the 5W1H questions. The answers to Who and Where questions are provided by the classification model, the answers to When question

is provided by the time series model, the answer to the What question is provided by the topic model, and the answer to How and Why will be provided by the question answering model. As we intended to use models from this study to create a StoryNet using questions answering system, we considered 5W1H classification as a natural language understanding (NLU) problem and designed BERT models of classification [117] to answer Who and Where questions.

We could also have use the question answering model for this category, but because we know the domain, we are using classification model which proved to improve the accuracy better than the question answering model. In case of a new domain we can still use the question answering model for all the questions to get the answers instead of using different models for different questions. Just for the sake of demonstrating the possibilities, we decided to show the best possible case / scenario that can be achieved by using this framework.

BERT is Bidirectional Encoder Representation from Transformers that relies entirely on self-attention instead of sequence-aligned recurrent neural networks (RNNs) or convolutions. It consists of multiple bidirectional transformer encoder layers [100]. Each layer, surrounded by a residual connection, has a multi-head self-attention mechanism, followed by a position-wise fully connected feed-forward network. An attention function can map a query and a set of key-value pairs to an output.

The output is a weighted sum of the values, and the weight assigned to each value is a compatibility function of the query with the corresponding key. The attention weights are calculated by Equation 3.1: the three inputs are  $Q$  queries,  $K$  keys, and  $V$  values;

and the output is the softmax of standard dot-product attention,  $QK^T$  of  $Q$  and  $K$  ( $K^T$  represents the transpose of matrix  $K$ ) with a scaling factor of  $\sqrt{d_k}$ , where  $d_k$  is the dimension of the key, ensuring the value of the dot product does not grow too large.

$$Attention(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad (3.1)$$

The attention scores  $e_{ij}$  are calculated by Equations (1), and (2).  $d_e$  is the output dimension, and  $W_Q$  (query),  $W_K$  (keys), and  $W_V$  (values) are the parameter matrices.

$$a_{ij} = \frac{(h_i W_Q)(h_j W_K)^T}{\sqrt{d_e}} \quad (3.2)$$

$$e_{ij} = \frac{exp\ e_{ij}}{\sum_{K=1}^N exp\ e_{ik}} \quad (3.3)$$

The output is calculated using Equation (3). It is the weighted sum of the previous outputs  $h$  and  $h_i$ .

$$o_i = h_i + \sum_{j=1}^N e_{ij}(h_j W_V) \quad (3.4)$$

Our classification model includes two parts: a BERT encoder that encodes 311 records/descriptions and a classification decoder that classifies a 311 call record into a 311 category (BERT-C) or a service department (BERT-D). The two classification tasks in problem and solution domains are trained separately.

We used the BERT encoder transformer with an added layer of classification decoder, similar to Next Sentence classification. The encoding of a text description is described in Equation 3.5.  $x_i$  is the representation of each token; and,  $h_i$  is the contextual semantic representation embedding of a token. Thus,  $H = (h_1, \dots, h_T)$ , the encoder outputs are the semantic representations of each record.

$$H = BERT(x_1, \dots, x_T) \quad (3.5)$$

Given an input token sequence  $X = (x_1, \dots, x_T)$ , the output of the BERT encoder is  $H = (h_1, \dots, h_T)$ , and  $h_i$  is the averaged output from the multi-headed transformer blocks given as token's contextual semantic representation embedding.

The hidden representation  $H \in R^{|X|h}$  is obtained by  $H = \text{BERT}(X)$ , where  $|X|$  is the length of the input sequence  $X = (x_1, \dots, x_T)$  and  $h$  is the size of the hidden dimension. Then,  $H$  is passed to a dense layer  $W \in R^{h|V|}$ , followed by softmax, as described in Equation 3.6. The classification decoder uses sentence semantic representation  $H$  to predict the class label  $y^c$ :

$$y^c = \text{softmax}(WH + b) \quad (3.6)$$

$y^c$  gives the prediction to the answers for Who and Where questions. Softmax is an activation function that converts a vector of numbers into probabilities within the range of  $[0, 1]$ :

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (3.7)$$

$\vec{z}$  is input vectors.  $e^{z_i}$  is the standard exponential function for input vectors.  $K$  is the number of classes.  $e^{z_j}$  is the standard exponential function for output vectors.

Given an input token sequence  $x = (x_1, \dots, x_T)$ , the output of the BERT encoder is  $H = (h_1, \dots, h_T)$ . The 5W1H decoder uses sentence semantic representation  $H'$  to predict the class label  $y^{intent}$ :  $y^{intent} = \text{softmax}(W^{int}H' + b^{int})$ .

Second, the input tokens to the slot filling decoder is  $h_i''$ , which joins the BERT encoder and the classification decoder with the slot filling decoder. Specifically  $h_i''$  is the concatenation of the BERT output token embeddings  $h_i$ , and the classification decoder token embedding  $h'$ . The slot filling decoder inputs the hidden states to a softmax layer



to predict the slot tags:  $y_i^{slot} = \text{softmax}(W^{slot}h_i^s + b^{slot})$ .

Using BERT for a specific task is relatively straightforward: BERT can be used for a wide variety of language tasks while only adding a small layer to the core model as shown in Figure 10. In the training process, the pairs of sentences as input are processed and learns to predict if the second sentence in the pair is the subsequent sentence in the original document.

During training, 50% of the inputs are a pair in which the second sentence is the following sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence. To help the model distinguish between the two sentences in training, the input is processed in the following way before entering the model.

A [CLS] token is inserted at the beginning of the first sentence, and a [SEP] token is inserted at the end of each sentence. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2. A positional embedding is added to each token to indicate its position in the sequence. The idea and implementation of positional embedding are presented in the Transformer paper [56].

Classification tasks are done similarly to Next Sentence classification by adding a classification layer on top of the Transformer output for the token. In Question Answering tasks (e.g., SQuAD v1.1), the software receives a question regarding a text sequence and is required to mark the answer in the series. Using BERT, a Q&A model can be trained

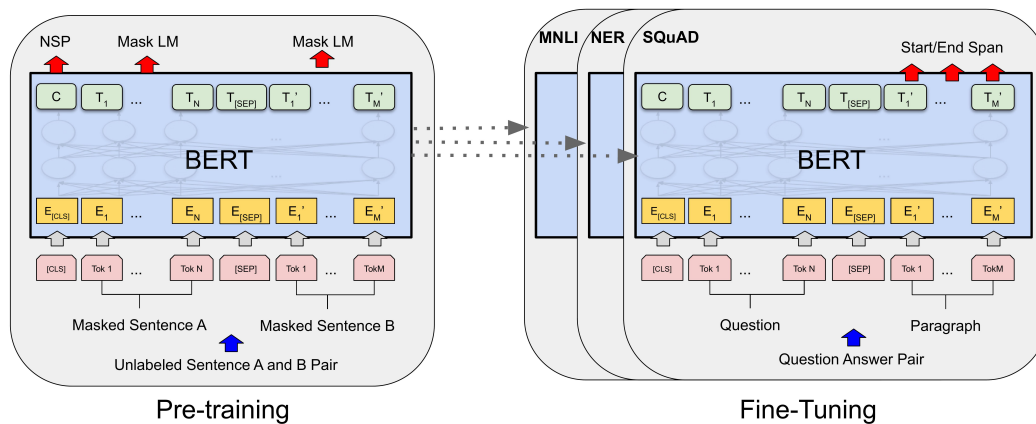


Figure 10: BERT - Fine Tuning

by learning two different vectors that mark the beginning and the end of the solution. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organization, Date, etc) that appear in the text. Using BERT, a NER model can be trained by feeding each token's output vector into a classification layer that predicts the NER label.

Using this question answer model for getting answers to 5W1H questions allows us to train the model using Electra. The architecture of the model is presented in Figure 11. Annotations generated from the aforementioned models and the models discussed in the latest sections, are used as input for this model which encompasses all the 5W1H questions.

### 3.2 “What” Model: LDA+BERT Custom Topic Model

As stated in the second problem, we need a model to categorize residents' complaints inductively and connect them with the internal-facing 311 service categories. The

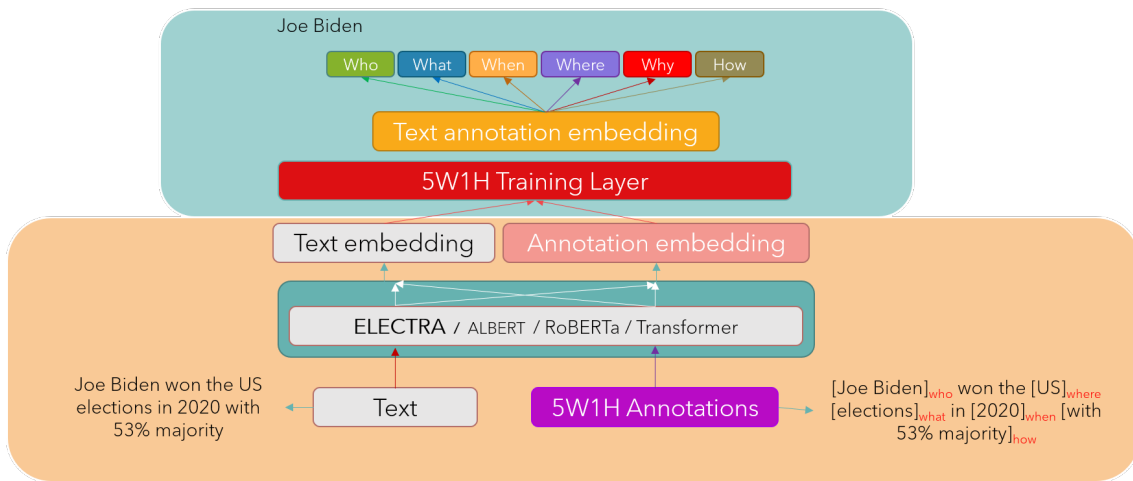


Figure 11: 5W1H-StoryNet Model

internal-facing 311 service categories are sometimes semantically arbitrary and thus hard for the external audience to understand.

“why” & “how.” Unsupervised method is more appropriate than supervised method to explore “why” & “how” as these two aspects of the 5W1H frame is highly idiosyncratic, depending upon each record and domain knowledge. The unsupervised inductive clustering creates opportunities for classification without any prior, and thus allows infinite classifications in real-world event classification. We’ll discuss more about this in question answering section.

Topic modeling has been widely used for analyzing domain-specific perspectives. Latent Dirichlet Allocation (LDA) [10] is one of the most popular approaches. To solve this problem, we make use of a custom topic modeling, with LDA+BERT Clustering, by combining LDA [10] and BERT [24]: LDA [10] is first used to detect topic per document probabilities, which is then combined with BERT [24] sentence embedding through

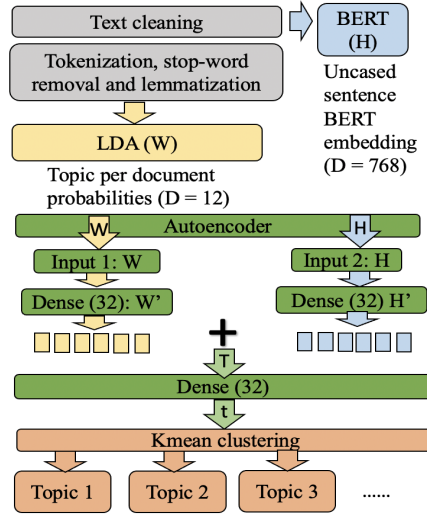


Figure 12: Architecture of LDA+BERT Clustering

an autoencoder. Finally, the latent representations from the encoder are entered into a clustering algorithm to categorically cluster documents. The pre-trained uncased BERT is used to generate document representation  $H$ . The LDA document vector  $W$  and the BERT document vector  $H$  are entered into an autoencoder.

Our custom topic model is different from the existing LDA+BERT model, since it balances the LDA topic and BERT document vectors. We came to this design via experimenting a few different structures, including direct concatenation before auto encoder, imbalanced LDA topic vector and BERT vector, different length of LDA topic vectors, and different number of clusters. The details of the experimentation are included in Case Study. Please refer to Figure 12 for the overview of architectural components.

We first employ uncased BERT, a pre-trained language model, to obtain the document representation  $H$  (Equation 3.1 and Equation 3.5). This model aims to identify the topic that the model would develop and plot a corresponding word cloud for each of

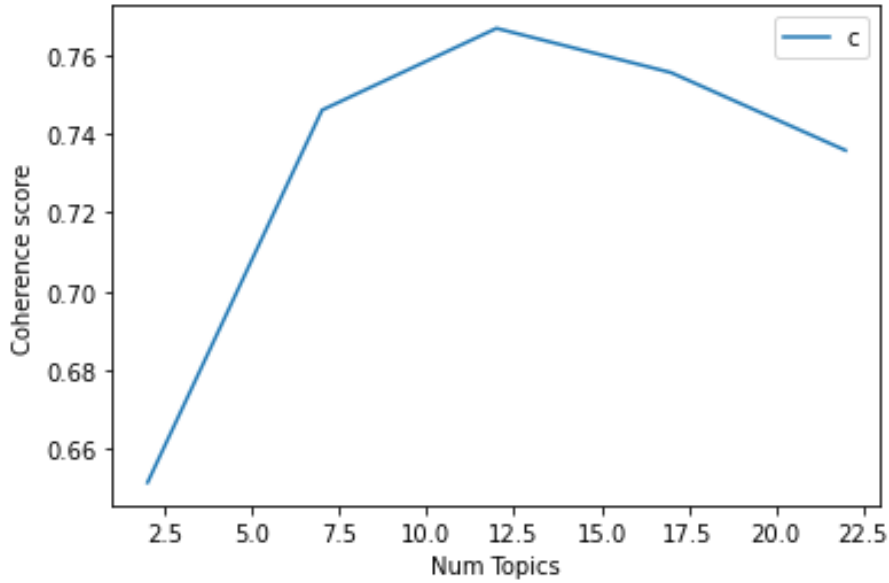


Figure 13: Topic Coherence

the  $K$  unique categories where the number of issues  $K$  was determined by the optimal topic model based on KL-divergence which signifies the coherence score of the topics as shown in Figure 13. The dominant topic for each description is determined by comparing the probabilities that the description belongs to the topics and is identified in the dataframe as shown in Figure 14.

**Pre-trained language models:** As our model is based on pre-trained BERT, we briefly describe the BERT model here. The BERT document vector  $H$  is generated using Equation 3.1 and Equation 3.5. The self-attention mechanism is described in Equation 3.1. Given an input document  $D$ , the uncased BERT model outputs the semantic representations of the document  $H$ , as described in Equation 3.5.

We intend to combine the strengths of LDA and BERT. The performance of the

Dominant_Topic	Topic_Perc_Contrib	Keywords	Text
Trash	0.1238	[[trash', 58.97], [front', 12.82], [row', 7.69], [noth', 5.13], [bull', 2.56], [shapiro', 2.56], [broadway', 2.56], [crosswalk', 2.56], [door', 2.56], [violate', 2.56]]	Citizen is reporting dumping of trash and large items at the curb is a nuisance and is spilling out on to the sidewalk. Citizen did not specify exactly the items that are being dumped.
Signs	0.0957	[[sign', 25.93], [recycl', 14.81], [car', 14.81], [traffic', 11.11], [locat', 11.11], [bvd', 7.41], [cater', 3.7], [hair', 3.7], [build', 3.7], [west', 3.7]]	Citizen reporting there is no stop signs either direction at 149th and Botts. Citizen reporting vehicles are just turning onto Botts from 149th and causing near accidents.
Property Violations	0.1274	[[property', 32.43], [attach', 13.51], [issu', 10.81], [grass', 10.81], [junk', 10.81], [truck', 8.11], [yesterday', 5.41], [abandon', 2.7], [mailbu', 2.7], [trailer', 2.7]]	There is a caucasian man building a lean-to behind the dumpster adjacent to my property. The dumpster is in the parking lot of what was 'Bonds Chicken and Blues', which is now vacant. I am concerned about safety and public health.
Street	0.1199	[[street', 38.24], [time', 17.65], [bin', 11.76], [peopl', 5.88], [bridg', 5.88], [supervisor', 5.88], [note', 5.88], [train', 2.94], [gregori', 2.94], [driveway', 2.94]]	Citizen is reporting several wrecked vehicles parked on the street, where school buses cannot safely navigate down the roadway due to all the wrecked vehicles in front of the business parked on the street.
Traffic	0.1271	[[traffic', 25.93], [recycl', 14.81], [car', 14.81], [sign', 11.11], [locat', 11.11], [bvd', 7.41], [cater', 3.7], [hair', 3.7], [build', 3.7], [west', 3.7]]	Citizen is reporting multiple vehicles that are unlicensed and are disabled and have not moved in weeks, that are parked on the street and preventing vehicular traffic from safely navigating thru the street.
Water	0.0994	[[water', 25.0], [pothol', 18.75], [yard', 12.5], [today', 9.38], [state', 9.38], [tenant', 6.25], [plate', 6.25], [hill', 6.25], [wagon', 3.12], [overgrowth', 3.12]]	Citizen is reporting various potholes on Water Works Rd before and after intersecting with M 9 Hwy.
Animals	0.1142	[[week', 19.23], [dog', 15.38], [window', 11.54], [hazard', 11.54], [bust', 11.54], [glass', 7.69], [jackson', 7.69], [amin', 7.69], [meyer', 3.85], [roof', 3.85]]	Citizen is a mail carrier and he reporting a large white pitbull loose in the area. Also a small sheppard mixed. These dogs are keeping him from delivering the mail. Pitbull aggressive, sheppard mix non-aggressive. Pitbull has chain. Sheppard mix has no collar. 67
Stickers	0.1143	[[month', 15.15], [block', 12.12], [vehicl', 12.12], [address', 9.09], [sticker', 9.09], [branch', 9.09], [meter', 9.09], [citr', 9.09], [park', 9.09], [emerg', 6.06]]	Citizen called TrashBot and reports their items not collected. It was out on time with no violation sticker.
Sidewalks	0.1121	[[lane', 29.41], [side', 14.71], [home', 14.71], [barri', 8.82], [sidewalk', 8.82], [chariot', 5.88], [barrier', 5.88], [week', 5.88], [trail', 2.94], [forward', 2.94]]	Citizen states that it smells like sewer outside the home. Specifically there is a small hole near her drive way but it is right before the sidewalk and the odor seems to be coming from there.
Parks & Recreation	0.1218	[[park', 20.69], [street', 13.79], [tree', 13.79], [intersect', 10.34], [hous', 10.34], [ter', 6.9], [taker', 6.9], [debr', 6.9], [trash', 6.9], [bull', 3.45]]	A large hole in the street by the guardrail of Roanoke Park. The hole is across the street from the north entrance to parking for Gordon Parks School.

Figure 14: Dominant Topic for 311 Record

LDA topic model could be influenced by the number of documents, the length of documents, and the number of topics [96]. Shorter texts, like 311 call records, may suffer from poor performance due to their length. This can be attributed to the LDA model’s random drawing of the document-topic, and topic-word proportion vectors [96].

While the length of 311 call records may undermine the performance of LDA, the coherence flow between sentences with each document may introduce opportunities to improve the performance of the topic model. Unlike tweets and short texts from social media, 311 call records are short. Still, each document may have a more coherent sentence flow since they are human-generated records of calls from residents about a specific instance. Here is an example of a 311 call record:

*“Citizen reports improper parking space striping. Codes state that spaces are 8.5 ft wide, but the spaces are only 8ft wide. The handicapped spaces are only 7.5 ft wide but are supposed to be 8.5 ft wide. The problem is likely to be present throughout the parking garage. Additionally, there should be handicapped parking signage in front of each stall, but there are none, just the logo on the ground.”*

We can see from this record that there is a clear semantic connection from sentence to sentence, and the entire record, due to high sentence-level coherence, is semantically focused. The same observation was made by Li et al. [52] on texts from Wikipedia. They proposed a bi-Directional Recurrent Attentional Topic Model (bi-RATM) for document embedding to capture sentence-to-sentence flow, and their model achieved state-of-the-art performance [52]. In the same spirit, our topic model uses BERT sentence level embedding to overcome the possible less-than-optimal performance of LDA on shorter texts (see

Figure 12).

**LDA topic vectors:** The LDA document vector is generated by Latent Dirichlet Allocation (LDA). It is a generative probabilistic model based on three primary structures, including words, topics, and documents [10]. The documents are represented as a random mixture over latent topics, and each topic is characterized by a distribution over words.

The generative process for each document  $w$  in a corpus  $D$  first choose  $N$  to be a Poisson distribution. Then, choose  $\theta$  to be a Dirichlet distribution. For each of the  $N$  words  $w_n$ , the generative process will choose a topic  $z_n$  for it from  $\text{Multinomial}(\theta)$ , and then choose a word  $w_n$  from  $p(w_n|z_n, \beta)$ , a multinomial probability given the topic  $z_n$ . A distribution over topics is determined by the documents over a distribution of words with each topic.

To generate a document, LDA firstly samples a document-specific multinomial distribution over topics from a Dirichlet distribution. Then it repeatedly samples the words from these topics. Given an input corpus  $D$  ( $d \in D$ ) with  $V$  unique words and  $M$  documents, each document  $d$  contains a sequence of  $N$  words  $d = \{W_1, W_2, \dots, W_N\}$ ,  $n \in \{1, 2, \dots, N\}$ . Given a topic number  $K$ ,  $k \in \{1, 2, \dots, K\}$ , the generative process will generate documents based upon per-document topic distribution and per-topic word distribution and optimize probabilities.  $\alpha$  is the per-document topic distribution. It is a matrix where each row is a document, and each column is a topic.

It indicates the likelihood that a document contains topic  $Z_k$ .  $\beta$  is the per-topic word distribution. This matrix has rows to represent topics and columns words, indicating



how likely a topic  $Z_k$ ,  $k$  contains word  $W_n$ .  $\theta_d$  is a multinomial distribution of documents drawn from a Dirichlet distribution with the parameter  $\alpha$ .  $\varphi_k$ ,  $k$  is a multinomial distribution of words in a topic drawn from a Dirichlet distribution with the parameter  $\beta$ . For each word position  $n$ , select a hidden topic  $Z_n$  from the multinomial distribution parameterized by  $\theta_d$ . And, then select  $W_n$  from  $\varphi_{Z_n}$ . If  $w$  is a document, consisting of a sequence of  $N$  words:  $w=(w_1, w_2, \dots, w_N)$ .  $D$  is a corpus of a collection of  $M$  documents.

Given a topic number  $K$ , the generative process first chooses  $\theta \text{ Dir}(\alpha)$  (Dirichlet distribution).  $\alpha$  is the per-document topic distribution. It is a matrix where each row is a document and each column is a topic, representing the likelihood that document  $d_i$  contains topic  $K_j$ . Then, for each of the  $N$  words  $w_n$ , choose a topic  $z_n \text{ multinomial}(\theta)$ . Then, choose a word  $w_n$  from  $p(w_n|Z_n, \beta)$ . choose a word  $w_n$  from  $p(w|\theta, \beta)$ .  $\beta$  is the per-topic word distribution. This matrix has rows to represent topics and columns words, indicating how likely the topic  $K_i$  contains word  $w_j$ . This process defines the marginal distribution of a document as a continuous mixture distribution. Thus, the word distribution is:

$$p(w_n|\theta, \beta) = \sum_z p(w_n|z, \beta)p(z|\theta) \quad (3.8)$$

$z$  is the topic for the  $n$ -th word, meaning  $w_n$  in document

$$p(w|\alpha, \beta) = \int p(\theta|\alpha) \left( \prod_{n=1}^N P(w_n|\theta, \beta) \right) d\theta \quad (3.9)$$

$\alpha$  is the per-document topic distribution. It is a matrix where each row is a document and each column is a topic, representing the likelihood that document  $d_i$  contains topic  $K_j$ .

$$P(W, Z, \theta, \varphi; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\varphi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \varphi_{j,t}) \quad (3.10)$$

The probability that a document will be generated is determined by Given an appropriate topic number  $K$ , the generative process for a document  $d$  is as following: Sample a  $K$ -vector  $\delta_d$  from the Dirichlet distribution  $p(\delta|\alpha)$ , where  $\delta_d$  is the topic mixture proportion of document  $d$ . For  $i = 1 \dots N_d$ , a sample word  $w_i$  in the document  $d$  from the document-special multinomial distribution  $p(w_n|\delta_d, \beta)$ , where  $\alpha$  is a  $k$ -vector of Dirichlet parameters, and the Dirichlet distribution  $p(\delta|\alpha)$   $\beta$  is a  $K \times V$  matrix of word probabilities, where  $\beta_{ij} = p(w_{j=1}|z_{i=1}), i = 0, 1, \dots, K; j = 0, 1, \dots, V$ .

**Autoencoder:** The LDA document vector  $W$  is defined below:

$$W_i = TopicModel([T_1, \dots, T_N]) \in R^t \quad (3.11)$$

Where,  $T_i$  is the probability of the document belonging to the  $i$ -th topic.  $N$  is the number of topics.  $W$  is Input 1 of the encoder  $E$ . A full-connected neural networks (NN) is employed to learn  $W'$ , vector representations of  $W$ .  $W'$  has the same dimension  $d$  as  $H'$ . Let  $W = \varphi_{Z_n}$ ,  $w_1$  is the weights, and  $b_1$  is the bias. The output is  $W'$ .

$$W' = RELU(w_1 \times W + b_1) \quad (3.12)$$

The BERT document vector  $H$  has a dimension of  $d' = 768$ . It is entered into the encoder  $E$  as Input 2. A full-connected NN learns  $H'$ , vector representations of  $H$ .  $H'$  has the same dimension  $d$  as  $W'$ .  $w_2$  is the weights, and  $b_2$  is the bias:

$$H' = RELU(w_2 \times H + b_2) \quad (3.13)$$

$W'$  and  $H'$  are concatenated into a single document vector  $T$ .

$$T = W'H' \quad (3.14)$$

A full-connected NN learns latent vector representations of  $T$ .  $w_3$  is the weights, and  $b_3$  is the bias:

$$TopicVectors(T) = softmax(w_3t + b_3) \quad (3.15)$$

The decoder  $D$  mirrors the dimensions and layers of the encoder  $E$ .

**Clustering:** The vector representations of each document is the hidden state of the autoencoder.  $E$  is the encoder, and  $D$  is a document:

$$t_i = E(D_i) \quad (3.16)$$

Clusters of the documents, where each cluster indicates a topic category, are generated using a clustering algorithm like K-means. The map of the clusters is made using UMAP projection as shown in Figure 15.

The LDA document vector  $W$  is generated by Latent Dirichlet Allocation (LDA). It is a generative probabilistic model based on three primary structures, including words, topics, and documents [10]. The documents are represented as random mixture over latent topics, and each topic is characterized by a distribution over words. Given an input corpus  $D$  ( $d \in D$ ) with  $V$  unique words and  $M$  documents, each document  $d$  contains a sequence of  $N$  words  $d = \{W_1, W_2, \dots, W_N\}$ ,  $n \in \{1, 2, \dots, N\}$ . Given a topic number  $K$ ,  $k \in \{1, 2, \dots, K\}$ , the generative process will generate documents based upon per-document topic distribution and per-topic word distribution, and optimize probabilities.

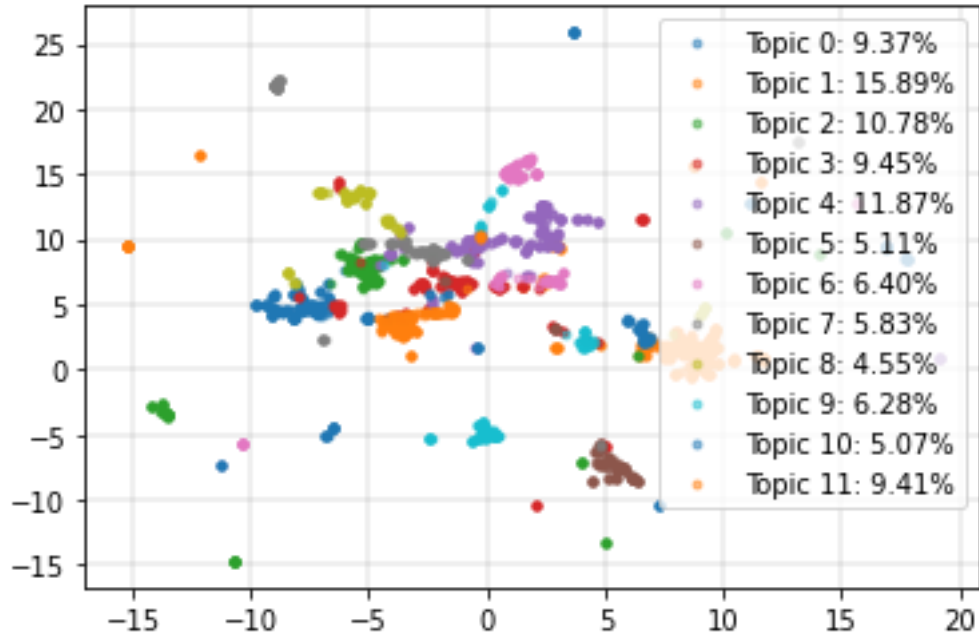


Figure 15: Topic Clusters - UMAP

$\alpha$  is the per-document topic distribution. It is a matrix where each row is a document and each column is a topic. It indicates the likelihood that a document contains topic  $Z_k$ ,  $k \in \{1, 2, \dots, K\}$ .  $\beta$  is the per-topic word distribution. This matrix has rows to represent topics and columns words, indicating how likely a topic  $Z_k$ ,  $k$  contains word  $W_n$ .  $\theta_d$  is a multinomial distribution of documents drawn from a Dirichlet distribution with the parameter  $\alpha$ .  $\varphi_k$ ,  $k$  is a multinomial distribution of words in a topic drawn from a Dirichlet distribution with the parameter  $\beta$ . For each word position  $n$ ,  $n \in \{1, 2, \dots, N\}$ , select a hidden topic  $Z_{-n}$  from the multinomial distribution parameterized by  $\theta_d$ . And, then select  $W_{-n}$  from  $\varphi_{Z_{-n}}$ . A visual representation of the topics in each document is shown in Figure 16.

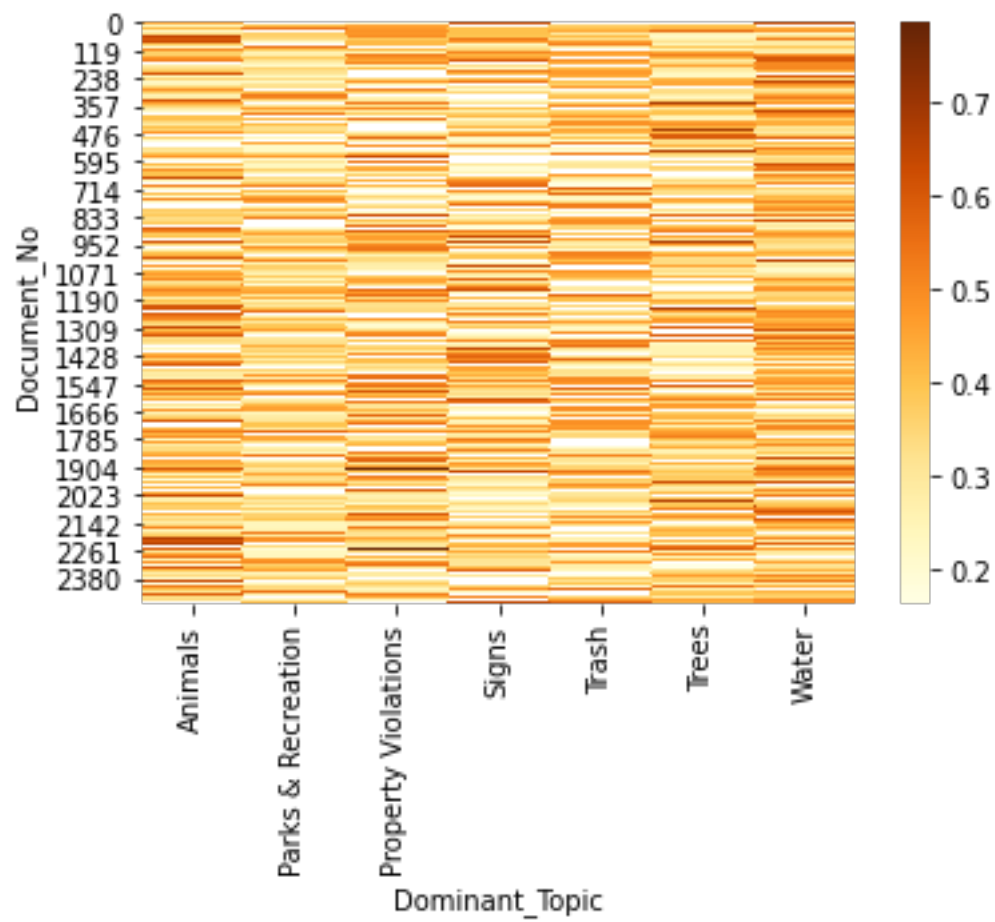


Figure 16: Topic Distribution in Documents

### 3.3 Objective 1. 5W1H component detection

Let us examine the process of detecting 5W1H components using the auto annotation scheme as shown in Figure 17. The BERT encoder transformer in “Who” and “Where” Model is used to create semantic representation of  $H = (h_1, \dots, h_T)$ , which  $X = \{x_1, \dots, x_T\}$  is a collective word vector of word embedding, segment embedding, and position embedding of each token in the 311 record. We used the uncased pre-trained sentence BERT [89]. The preprocessing of LDA topic modeling includes text cleaning, tokenization, stop-word removal, and lemmatization. Then, coherence score analysis is used to determine the optimal number of topics. The LDA model then generates topic per document probabilities, i.e., the LDA document vector  $W$ . The preprocessing of BERT document embedding is text cleaning. Then, the pre-trained uncased BERT is used to generate document representation  $H$ .

The LDA document vector  $W$  and the BERT document vector  $H$  are entered into an autoencoder. The LDA document vector  $W$  is entered as Input 1 and the dimension is increased to 32 via a full-connected NN. The output is  $W'$ . The BERT document vector  $H$  has a dimension of 768. It is entered as Input 2, and the dimension is reduced to 32 via a full-connected NN. The output is  $H'$ .  $W'$  and  $H'$  are concatenated into a single document vector  $T$ , and reduced to 32 via a dense layer to get  $t$ .  $H$  is concatenated with the LDA document vector  $W$ .

All token in a 311 call record,  $X = \{x_1, \dots, x_T\}$ , are passed to the LDA topic model using delta ( $\delta$ ) hyper-parameter to add relative importance to infer topic distribution

per token ( $W_i$ ):

$$W_i = TopicModel([T_1, \dots, T_N]) \in R^t TopicVectors(T) = Wt + H \quad (3.17)$$

Then, the combined vectors  $t$  are entered into a simple autoencoder with a dense layer, where the latent vector space utilizes dimensionality reduction and noise to determine the topic clusters. Finally, clusters of the documents, where each cluster indicates a topic category, are generated using a clustering algorithm like  $s$  (see Figure 12). This design is different from the existing LDA+BERT model, since it balances the LDA topic and BERT document vectors. We came to this design via experimenting a few different structures, including direct concatenation before auto encoder, imbalanced LDA topic vector and BERT vector, different length of LDA topic vectors, and different number of clusters. The details are included in Case Study.

In order to obtain the contextual topics, the topic vectors ( $\omega$ ) of LDA model are merged with a collective contextual word embedding vectors ( $H$ ) from Sentence-BERT model using a gamma ( $\gamma$ ) hyper-parameter to add relative importance to both vectors as shown in (2).  $H = BERT(x_1, \dots, x_T)$  Contextual Topic Vectors( $t$ ) =  $\omega\gamma + H$  (2) where  $x_1, \dots, x_T$  is a collective word vector of word embedding, segment embedding, and position embedding of each tweet token; Trm stands for Transformer encoder unit;  $H = (w_1, \dots, w_T)$ , and  $w_i$  the averaged output from 12 multi-headed transformer blocks given as token's contextual embedding vector representation.

The combined vectors( $t$ ) are passed into a deep learning auto-encoder latent vector space to ensure dimensionality reduction and noise to arrive at the best topic clusters. The output of the auto-encoder is a cluster of keywords, each falling into a specific unique

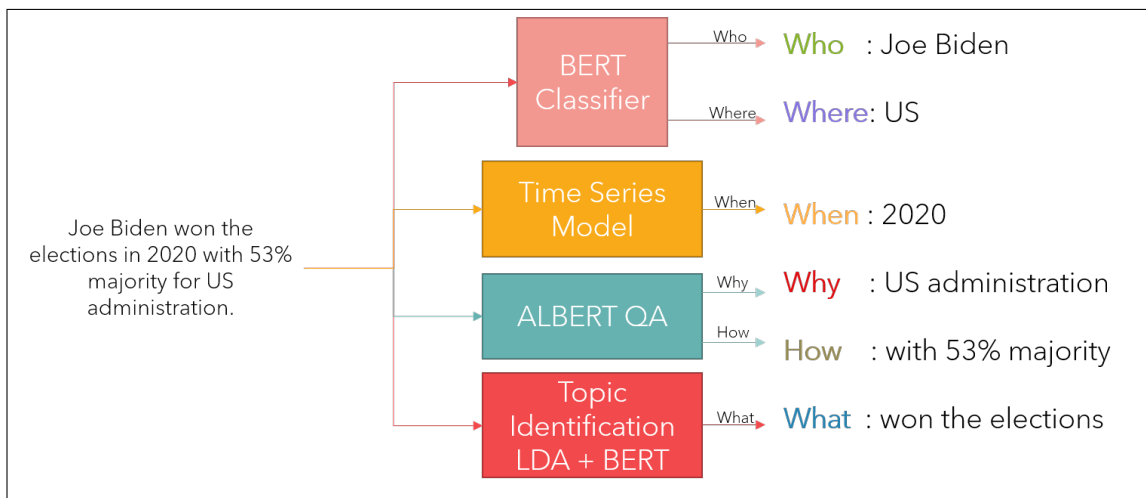


Figure 17: Auto Annotation

topic category using a clustering algorithm are labeled manually with unique topics. This approach is expected to provide more accurate topic semantic information for simulation on short tweet texts.

In LDA, the topic proportions are randomly drawn from a Dirichlet distribution, which implies the independence between topics. However, these correlations are widespread in real-world data. Interestingly, the association or correlation of topics can be modeled with LDA. The topic “311 services” is often correlated with “crimes” while unlikely co-occurs with “business.” There would be an inconsistency between the assumption and input documents so that the predefined parameter  $K$  may not be able to reflect the real topics of the domain. To overcome the limitation, the NLP embeddings, such as BERT, could be significantly contributed to determining relevant or non-related topics in the 311 call domain.

Following this, LDA+BERT embedding is used to inductively explore event-specific



frame elements, namely “why” & “how.” Unsupervised method is more appropriate than supervised method to explore “why” & “how” as these two aspects of the 5W1H frame is highly idiosyncratic, depending upon each call and caller’s knowledge about the situation. The unsupervised inductive clustering creates opportunities for classification without any prior, and thus allows infinite classifications in real-world event classification.

Topic modeling has been widely used for analyzing domain-specific perspectives. Latent Dirichlet Allocation (LDA) [10] is one of the most popular approaches. Advanced LDA-based topic modeling approaches [59, 99] were proposed for the question and answering community. JAIST [99] combined multiple features for question-answering community websites like Yahoo ([77], [99], [59], [111]).

And, topic modeling has been incorporated with contextual language modelling [32] and machine translation [18], summarization [64, 106]. In this thesis, we are interested in exploring the perspectives of the topics based on the contextual topic embedding (LDA + BERT). The word clouds for the identified topics have been given in Chapter 4.

### **3.4 “When” Model: Time Series Forecasting**

The Prophet model was used to predict the estimated responding time for a specific 311 service call using the 311 service data from the past ten years. The Prophet model is a Generalized Additive Model (GAM) [98]. The aspect of “when” is handled by our time series forecast model. The predictions from the classification models for departments and 311 service categories were used to train time-series prediction.

The Prophet model was built to predict the estimated responding time for a specific

311 service call using the 311 service data from the past ten years. The major components of Prophet model included growth forecasting, a model to understand how the population has grown and will be continuously growing. The Prophet model includes three major components:  $g(t)$  is the trend,  $s(t)$  is seasonality,  $h(t)$  is holidays, and  $\epsilon_t$  is the error term. They are summed up to perform forecast:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \quad (3.18)$$

$g(t)$  is modeled using the piece-wise logistic growth model as follows: The trend changes in the growth model have been adjusted by explicitly defining change points, where the growth rate is allowed to change. For the given  $S$  change points at times  $s_j, j = 1, \dots, S$ , a vector of rate adjustments is defined as  $\delta \in R^S$ , where  $\delta_j$  is the change in rate that occurs at time  $s_j$ .

The rate at any time  $t$  is then the base rate  $k$ , plus all of the adjustments up to that point:  $k + \sum_{j: s_j < t} \delta_j$ . A vector is defined as  $a(t) \in \{0, 1\}^S$  and the rate at time  $t$  is then  $k + a(t)^T \delta$ . When the rate  $k$  is adjusted, the offset parameter  $m$  is also adjusted to connect the endpoints of the segments. The correct adjustment  $\gamma$  at change point  $j$  is defined as follows:

$$g(t) = \frac{C(t)}{1 + \exp((k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \quad (3.19)$$

with  $C(t)$  is time-varying capacity,  $k$  the growth rate, and  $m$  an offset parameter. Also, in order to fit and forecast these effects, seasonality models have been defined as periodic functions of  $t$  using Fourier series considering periodic effects.

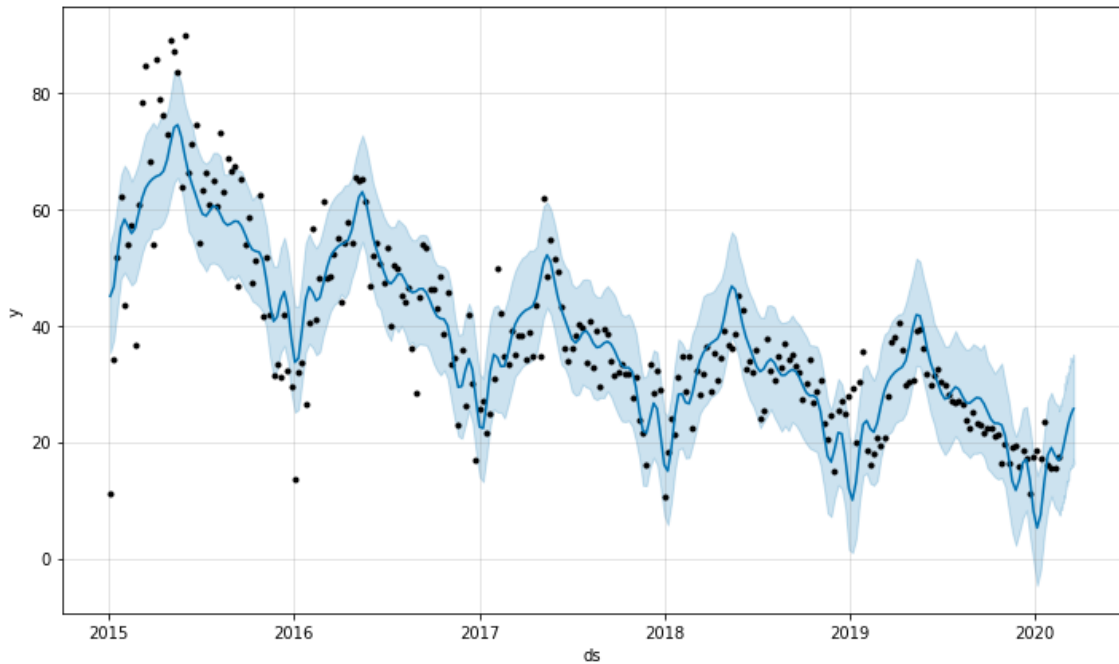


Figure 18: Time Series Analysis

We have also explored several deep learning models, including LSTM and CNN models, for the time series forecasting. Surprisingly, our case study showed the Prophet forecasting model had superior prediction compared to the deep learning models to be discussed later, suggesting city service delivery is influenced by seasonality and holidays.

**Data:** The time series forecasting was conducted on all the data available from 2015 to the current time. The results are shown in Figure 18. It shows a steadily decreasing trend overtimes, which suggests an improvement in the performance of the resolution team. The cases that were used to take around 80 days of resolution time in 2015 have been found to be solved in under 20 hours in 2020.

**Training:** To further drill down upon the trends of the case resolution, we have analyzed the same data based upon the category of the 311 service requests. The category-wise

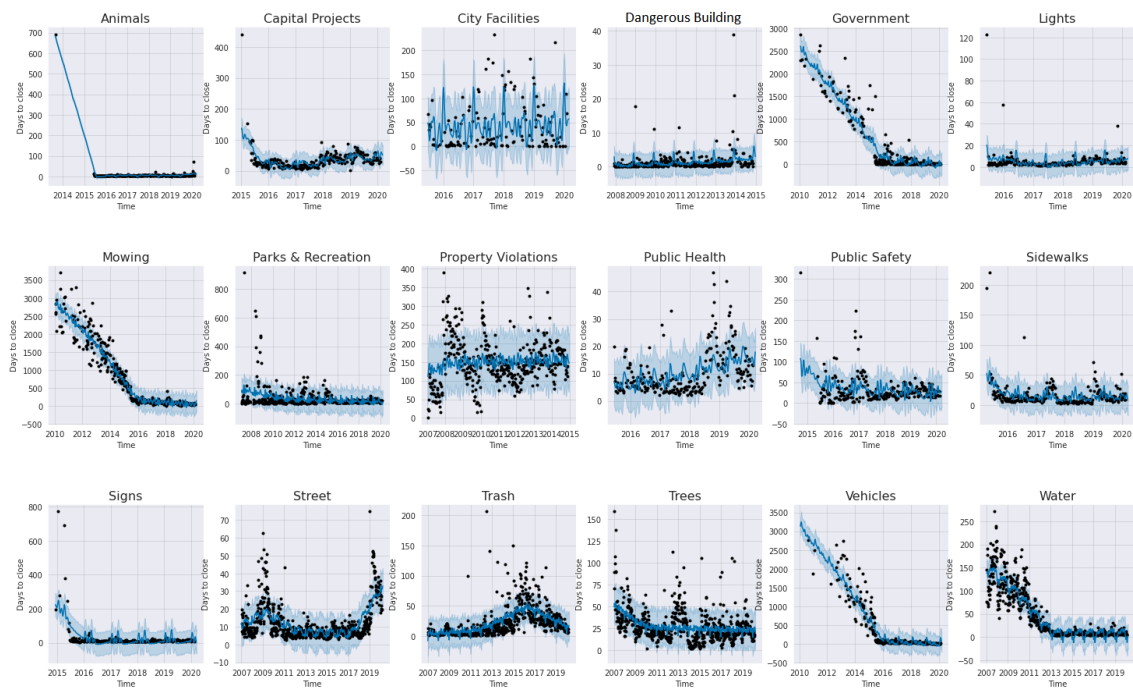


Figure 19: 311 Service Category Wise Time Prediction

trends have been shown in Figure 19.

**Evaluation:** Table 20 shows the evaluation of time series results on a 70-30 split of data. In conjunction with the overall trend, many of the categories exhibit the steady decrease of the resolution time (in days) over time from the beginning of the data availability, with *Public Health* being an exception. The issues related to *Trash* have been observed to rise during four years of 2014-2018, but have subsided after that. The issues related to *Lights*, *Animals*, *Capital Projects*, *Sidewalks*, *Trees*, *Animals* and *Property Violations* have been showing a steady trend. The steep decreasing trend for *Vehicles*, *Mowing*, *Government*, and *Water* issues shows that there have been very effective responses to the requests in Kansas City.

### TIME SERIES PREDICTION PER 311 CATEGORY

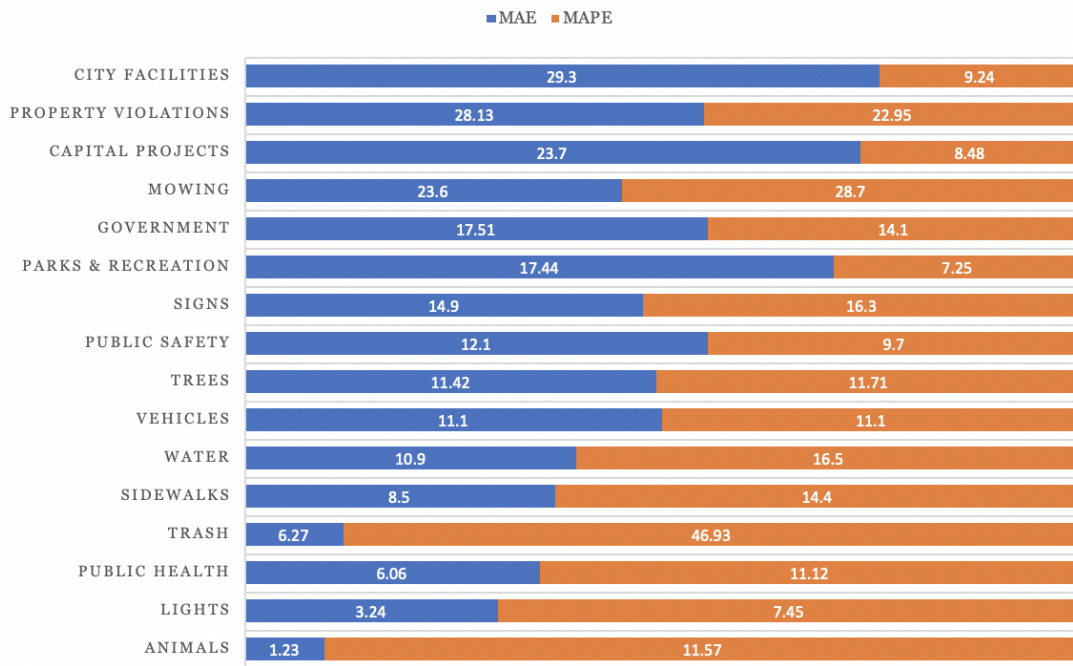


Figure 20: 311 Call Time Series Validation

The time series model for the 311 service data, which is a nonlinear regression based model for 311 call response times, was evaluated: (1) MAE (Mean Average Error) (defined in Eq. 3.20) is the measure of the difference of predicted versus observed, (2) MAPE (Mean Average Percentage Error) (defined in Eq. 3.21) is a measure of prediction accuracy of the forecasting (loss function for regression in machine learning).

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n} \quad (3.20)$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (3.21)$$

Figure 20 shows the MAE and MAPE scores as the performance of the 311 time series model. The best classes of the response time prediction are *Animal* and *Tracy* with 1.23% and 6.27% of its MAE. The worst class accuracy is *City Facility* with 29.3% of MAE. In terms of MAPE, the best class is *Lights* with 7.45% of MAPE, and the worst type is *Tracy*, with 46.93% of MAPE. The overall accuracy for the response time estimation is about 70%. The Prophet forecasting model showed superior performance when compared to the deep learning models (Table 3).

### 3.5 “Why” & “How” Model: ELECTRA BERT / ALBERT / RoBERTa / T5

**Question answering:** StoryNet is a great way to share existing stories and create new accounts. It is an effective instrument for co-creation and co-teaching in STEM education. The StoryNet is designed as a knowledge graph so that the underlying representations of the stories are W3C standard Resource Description Framework (RDF) and RDF Schema.

Table 3: 311 Time Series Predictive Model Accuracy

Model Name	MAE	MAPE
LSTM	9.71	105.63
LSTM + Window	7.81	79.33
LSTM + Time Steps	5.70	65.87
LSTM + Memory	7.44	84.50
Stacked LSTM + Memory	7.84	93.62
DCNN (SeriesNet)	14.20	210.32
Prophet (Ours)	<b>3.45</b>	<b>14.38</b>

The similarities of stories are determined by NLP and topic modeling techniques based on their common contexts.

The 5W+H questions will be mapped to NLP-based question, and the answer embedding and the story embedding will be mapped to SPARQL and Shapes Constraint Language (SHACL), which allows for a semantic query for retrieving and manipulating data stored in Resource Description Framework format. For the 311 Q/A model, we used ALBERT [47] in which the backbone is similar to the transformer encoder [100] with GELU nonlinearities [38] built in BERT. We follow the BERT notation conventions and denote the vocabulary embedding size as  $E$ , the number of encoder layers as  $L$ , and the hidden size as  $H$ .

Following [25], we set the feed-forward/filter size to be  $4H$  and the number of attention heads to be  $H/64$ . Our innovation is to use the predictive models such as BERT-311, the time series forecasting model, and the topic identification model to build the AI-Community 311 Q&A framework based on the ALBERT question answering. Our design of the Q&A follows the story framework presented earlier. We applied story framework

to the design thinking of answering each incoming 311 call.

Since one of BERT’s applications is in QA, different variations of BERT [16, 48, 53, 85] have been used. For the competition, many teams used BERT to develop QA systems [84, 113]. For instance, [113] used BERT to find relevant answers to keywords extracted from a question. The found solutions were ranked by *Universal Sentence Encoder Semantic Similarity* or USESS then *Bayesian Additive Regression Trees* or BART summarized the top results. Another team employed BERT as a semantic search engine to find answers.

The QA system produced semantically meaningful sentence embedding on the paragraph extracted from the CORON-19 dataset and found five paragraphs and their corresponding papers’ titles and abstracts. In [84], the QA systems were based on *A little BERT* or **ALBERT** [48] to find answers for questions related to COVID-19. Based on the above works, we develop QA systems using Transformers. To do so, we examined the performance of well-known transformers such as ALBERT, BERT, T5 Model. The next section explains the architectures and the results obtained by employing them on two datasets.

### 3.6 Transformer-Based QA Systems

For the competition, we aimed to develop a QA system using a high performance Transformer. To do so, we developed three QA systems using BERT-large, ALBERT-base and *Text-to-text transfer transformer* or (T5)-large, pre-trained them on various QA datasets and evaluated them on two labeled questions answers datasets —as described



below. To create answers to a query, we use three different transformer-based models pre-trained on various QA datasets, as described below.

### 3.6.1 Datasets for Pre-training Transformers

- We use various datasets to pre-train our QA systems:

**SQuAD v1.1** [81] —the *Stanford Question Answering Dataset* containing 100k question-answer pairs on more than 500 articles;

**SNLI** [12] —the *Stanford Natural Language Inference* corpus containing 570k human-written English sentence pairs manually labeled for balance classification with the labels entailment, contradiction and neutral;

**MultiNLI** [110] —the *Multi-Genre Natural Language Inference* corpus containing 433k crowd-sourced sentence pairs with the same format as SNLI except it includes a more diverse range of text and a test set for cross-genre transfer evaluation;

**STS** —the *Semantic Textual Similarity* benchmark is a careful selection of data from English STS shared tasks (2012-2017) comprising of 8.6k annotated examples of text from image captions, news headlines, and user forums; and

**Yahoo** —the question-answering dataset consists of questions with “exact” and “ideal” answers. We specifically use Yahoo factoid QA pairs, excluding yes/no or list QA pairs, because the factoid dataset has a similar structure as SQuAD v1.1 [81].

## 3.6.2 BERT-based

### 3.6.2.1 BERT-large

Our BERT-large QA system is developed using a pre-trained QA BERT-large-uncased model with whole word masking fine-tuned on SQuAD v1.1 [81]. The model contains 24 Transformer blocks, 1024 hidden layers, 16 self-attention heads adding up to 340M parameters in total.

### 3.6.2.2 ALBERT-base

The ALBERT-base QA system is formed using a pre-trained QA ALBERT-base-uncased model fine-tuned on SQuAD v1.1 [81]. The model contains 12 Transformer blocks, 768 hidden layers, 12 self-attention heads, adding up to 12M parameters in total.

## 3.6.3 T5-large

The T5-large QA system is based on the T5 model that is a modern, massive multitask model trained by uniting many NLP tasks in a unified text-to-text framework [80]. By leveraging extensive pre-training and transfer learning, it has achieved state-of-the-art performance on a variety of NLP benchmark tasks, including the GLUE benchmark [104]. Following work by [82], which explores the task of generative closed-book question answering, we explore the efficacy of generating (rather than extracting) COVID-19 answers directly from an input question, without context.

Unlike our preceding two approaches, the T5 model explores generation of answers to questions, without context. Using the pre-trained T5 model with 770M parameters released by [80], we fine-tune for 25000 steps on an equal-proportions mixture of three QA tasks using the Natural Questions dataset, Trivia QA dataset and the train split of the COVID-19 QA dataset [42,45]. Only the queries are given as input and answers are generated using simple greedy decoding. Evaluation is then performed on the test split of the dataset.

We emphasize that these results are not directly comparable to the other frameworks, as the model is faced with the challenging task of jointly localizing relevant information and then generating a coherent answer. The advantage of such a framework is that it is context-free, meaning that it requires the least data preparation and human intervention.

#### 3.6.4 RoBERTa

Introduced at Facebook, Robustly optimized BERT approach RoBERTa, is a re-training of BERT with improved training methodology, 1000% more data and compute power. To improve the training procedure, RoBERTa removes the Next Sentence Prediction (NSP) task from BERT's pre-training and introduces dynamic masking so that the masked token changes during the training epochs. Larger batch-training sizes were also found to be more useful in the training procedure.

Importantly, RoBERTa uses 160 GB of text for pre-training, including 16GB of

Books Corpus and English Wikipedia used in BERT. The additional data included CommonCrawl News dataset (63 million articles, 76 GB), Web text corpus (38 GB) and Stories from Common Crawl (31 GB). This coupled with whopping 1024 V100 Tesla GPU's running for a day, led to pre-training of RoBERTa. As a result, RoBERTa outperforms both BERT and XLNet on GLUE benchmark results.

On the other hand, to reduce the computational (training, prediction) times of BERT or related models, a natural choice is to use a smaller network to approximate the performance. There are many approaches that can be used to do this, including pruning, distillation and quantization, however, all of these result in lower prediction metrics.

### 3.6.5 ELECTRA

Efficiently Learning an Encoder that Classifies Token Replacements Accurately (ELECTRA) - is a novel pre-training method that outperforms existing techniques given the same compute budget. For example, ELECTRA matches the performance of RoBERTa and XLNet on the GLUE natural language understanding benchmark when using less than  $\frac{1}{4}$ th of their compute and achieves state-of-the-art results on the SQuAD question answering benchmark.

ELECTRA's excellent efficiency means it works well even at small scale. So, it can be trained in a few days on a single GPU to better accuracy than GPT, a model that uses over 30x more compute. ELECTRA is being released as an open-source model on top of TensorFlow and includes a number of ready-to-use pre-trained language representation models.

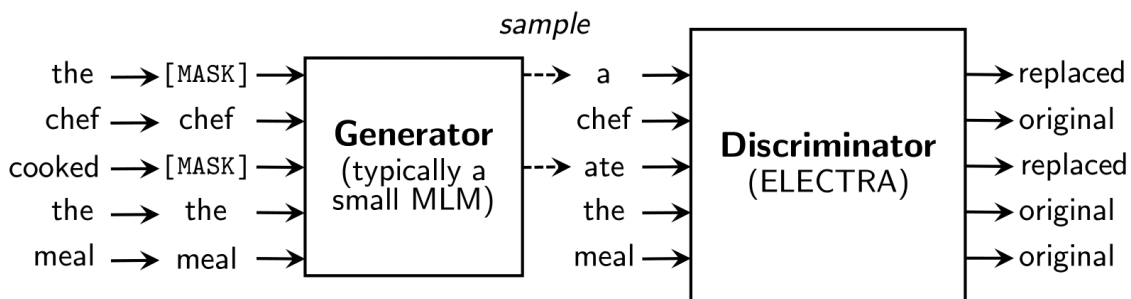


Figure 21: ELECTRA Architecture

The architecture of ELECTRA is specified in Figure 21. ELECTRA uses a new pre-training task, called replaced token detection (RTD), that trains a bidirectional model (like a MLM) while learning from all input positions (like a LM). Inspired by generative adversarial networks (GANs), ELECTRA trains the model to distinguish between “real” and “fake” input data. Instead of corrupting the input by replacing tokens with “[MASK]” as in BERT, our approach corrupts the input by replacing some input tokens with incorrect, but somewhat plausible, fakes. For example, the word “cooked” could be replaced with “ate”. While this makes a bit of sense, it doesn’t fit as well with the entire context.

The pre-training task requires the model (i.e., the discriminator) to then determine which tokens from the original input have been replaced or kept the same. Crucially, this binary classification task is applied to every input token, instead of only a small number of masked tokens (15% in the case of BERT-style models), making RTD more efficient than MLM - ELECTRA needs to see fewer examples to achieve the same performance because it receives more training signal per example. At the same time, RTD results in powerful representation learning, because the model must learn an accurate representation of the data distribution in order to solve the task.



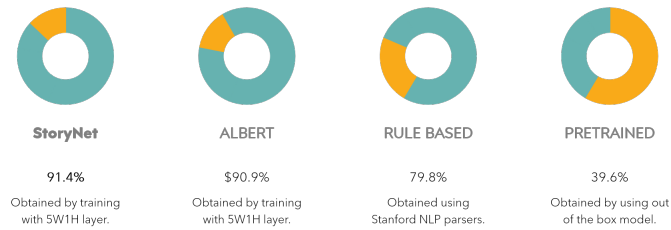


Figure 23: Performance Comparison - Models

provided in Chapter 4.

### 3.6.6 Performance Comparison

A consolidated comparison of the results for the studied models is shown in Figure 23. From the figure it can be inferred that our model has outperformed the ALBERT, Pre-trained BERT and the rule based models by a significant margin in terms of the ensemble score obtained using the metrics presented before.

Especially, when compared to the pretrained model, there is a significant jump from 39.6% to 91.4%. Thus the efficiency of the 5W1H layer, is very much key to the improvement of the StoryNet model. A more detailed analysis and breakdown of the scores is presented in Chapter 4.

## 3.7 Objective 2: Graph Generation

The 5W1H answers extracted by us so far, need to be connected together using an algorithm to create the StoryNet. In this section we make use of the answers were that we found out using the methods of question answering, topic modeling, classification as seen above. This process can be understood from the illustration in Figure 24. The individual

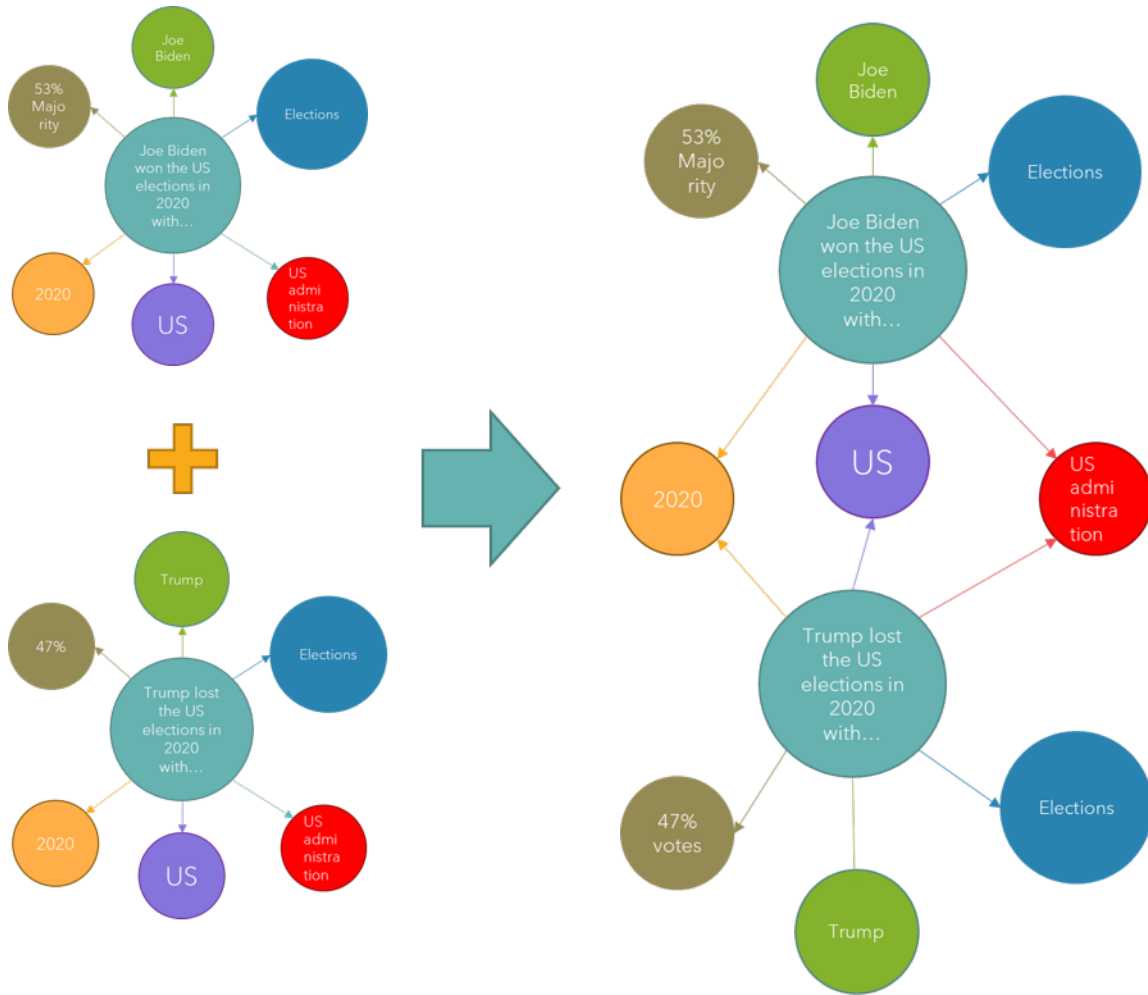


Figure 24: Graph Generation



```

MST-PRIM( $G, w, r$ )
1  for each  $u \in G.V$ 
2       $u.key = \infty$ 
3       $u.\pi = \text{NIL}$ 
4   $r.key = 0$ 
5   $Q = G.V$ 
6  while  $Q \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(Q)$ 
8      for each  $v \in G.Adj[u]$ 
9          if  $v \in Q$  and  $w(u, v) < v.key$ 
10              $v.\pi = u$ 
11              $v.key = w(u, v)$ 

```

Figure 25: Graph Connection Algorithm

stories with the 5W1H components as shown on the left side, are connected and match together using the following algorithm as shown in Figure 25 generate the graph that is more compact and connected as shown on the right side. An example for this graph using 311 call is shown in Figure 22. The last objective (3) of StoryNet application is discussed in the following chapter.

The nodes are the 5W1H components, which are connected to a story and these components are connected together using an algorithm as shown in Figure 25. It's a greedy algorithm that finds the MST (minimum spanning tree) for a weighted, undirected graph. Starting at an arbitrary vertex, the algorithm builds the MST one vertex at a time where each vertex takes the shortest path from the root node. Here is a description of the algorithm:

### 3.7.0.1 Algorithm to Combine Stories Graphs:

1. An empty set is created to keep track of vertices already included in the graph.
2. A key value is assigned to all vertices in the input graph. Initialize all key values as INFINITE. For the first vertex, a key value of 0 is assigned so that it is picked first.
3. While the set doesn't include all vertices:
  - (a) Pick a vertex  $u$  which is not there in the set and has minimum key value.
  - (b)  $u$  is included in the set.
  - (c) Key values of all the vertices adjacent to  $u$  are updated. To update the key values, iterate through all adjacent vertices. For every adjacent vertex  $v$ , if weight of edge  $u-v$  is less than the previous key value of  $v$ , update the key value as weight of  $u-v$ .

The idea of using key values is to pick the minimum weight edge from cut. The key values are used only for vertices which are not yet included in the graph, the key value for these vertices indicate the minimum weight edges connecting them to the set of vertices included in the graph.

## CHAPTER 4

### STORYNET APPLICATION: EVALUATION AND RESULTS

#### 4.1 Case Study 1: 311 Calls

##### 4.1.1 311 data

The domain of our study is the open dataset 311 of the Kansas City metropolitan area (from 2015 to 2020). To comprehend the reach of the data set in terms of decision-making, we performed static data analysis. Our analysis is based on the publicly available 311 open service data sets. The 311 call is a service provided by the government of the USA to deal with nuisance updates. Individuals can contact the 311 services by phone, on the website, or from Twitter to report any nuisance cases in their neighborhoods. These cases are updated daily on KCMO 311 service [43] - a repository of publicly available data.

This data set is updated almost in real-time, which allows us to analyze real-time problems identified by the community. The 311 service requests are usually channeled through multiple sources, including phone calls, emails, and social media. It should also be noted that the data set has a geographical signature of the incidents, which will allow the machine to comprehend location information of neighborhoods. The description of the incidents is, however, not available within the data set. This requires extraction using web scraping techniques. Web scraping is limited by the number of requests we could make to the hosting provider; we had to extract information on a timed basis.

The 311 service requests are usually channeled through multiple sources, including phone calls, emails, and social media. It should also be noted that the dataset has a geographical signature of the incidents, which will allow the machine to comprehend location information of neighborhoods. The description of the incidents is, however, not available within the dataset. This requires extraction using web scraping techniques. Web scraping is limited by the number of requests we could make to the hosting provider; we had to extract information on a timed basis. The five-year data from the 311 service data (from 2015 to the current time) are used to explore different dimensions of “what happened,” like neighborhood activities, violence, friendliness, maintenance, service departments’ responsiveness. Some of the key fields from the data are:

- Departments of 311 services,
- Categories of 311 services,
- Descriptions of the incidents,
- Geo-location coordinates of the incidents,
- Dates of the 311 service requests.

Figure 26 shows examples of a few records from the 311 dataset.

The 311 calls can be reported through multiple communication channels such as phone calls, email, Web, or social media. Most of them (about 97%) were reported by the three channels of phone (approximately 70%), Web (about 20%), and email (about 6.7%).

### 311 Call Center Service Requests: 2007 - March 2021

This data set contains service request data from the 311 call center in Kansas City, MO. In March 2021, Kansas City began transitioning to a

CASE ID	SOURCE	DEPARTMENT	WORK GROUP	REQUEST TYPE	CATEGO...	TYPE
2021033686	PHONE	NHS	NHS-Dangerous Buildings-	Prop/Build/Construct-Dangero...	Property / Bu...	Dangerous B...
2021033407	PHONE	NHS	NHS-Neighborhood Prese...	Property Violations	Property / Bu...	Property Mai...
2021033233	PHONE	NHS	NHS-Dangerous Buildings-	Prop/Build/Construct-Dangero...	Property / Bu...	Dangerous B...
2021033140	PHONE	NHS	NHS-Dangerous Buildings-	Prop/Build/Construct-Dangero...	Property / Bu...	Dangerous B...
2021032869	PHONE	NHS	NHS-Dangerous Buildings-	Prop/Build/Construct-Dangero...	Property / Bu...	Dangerous B...
2021032828	PHONE	NHS	NHS-Neighborhood Prese...	Property Violations	Property / Bu...	Property Mai...
2021032612	PHONE	NHS	NHS-Dangerous Buildings-	Prop/Build/Construct-Dangero...	Property / Bu...	Dangerous B...

Figure 26: 311 Data

The KCMO 311 service data [43] (shown in Table 4) is split into 80-20 train-validation ratio to train and evaluate the model’s performance. As seen in Figures 37 and 38, the total numbers of the internal-facing 311 service categories and departments are 17 (not including the “other” and “no data available” categories) and 15, respectively.

Table 4: KCMO 311 Service Request Dataset

Department#	Category#	Training#	Testing#	Total
15	17	112,412	28,103	140,515

As the 311 call data have spatial and temporal features, the visualization of 311 calls spatially and temporally would be utilized to discover valuable patterns. When paired with the power of charts, it becomes a powerful tool to derive insights on both analytical and semantic dimensions. The combined analysis of structured, unstructured, and spatial and temporal data proved to be an efficient way to derive a holistic picture of the state of the geographic entity under study (state/neighborhood / block-group).

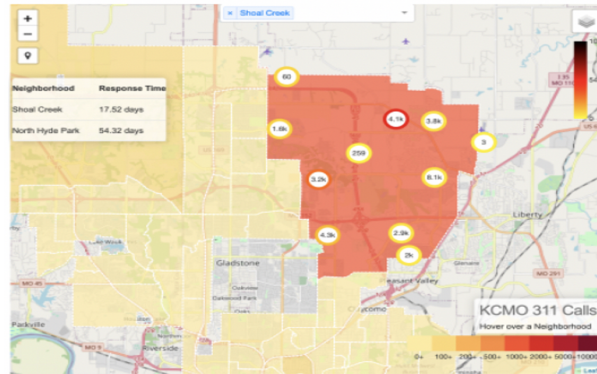


Figure 27: Visualization for 311 Call Map with Spatial Partition

#### 4.1.2 Map Visualization

Since the nature of call data allowed access to the location where the complaint/request is reported, we made use of this data to plot the requests on a map and analyze them using spatial hierarchical clustering (as shown in Figure 27). It uses a greedy hierarchical clustering algorithm to identify the center of a cluster and accumulates the volume of calls at a particular zoom level. As the zoom level increases (more granular), the locations of the calls are more distinct. Since we are dealing with spatial data, visualization of 311 calls spatially is an ideal means of identifying valuable patterns. Moreover, when paired with the power of charts, it becomes an effective tool in deriving insights on both analytical and semantic dimensions. For example, the combined analysis of structured, unstructured, and spatial data proved to be an efficient way to derive a holistic picture of the state of the geographic entity under study (state/neighborhood/block-group). An overview of these marked on a map is shown in Figure 28.

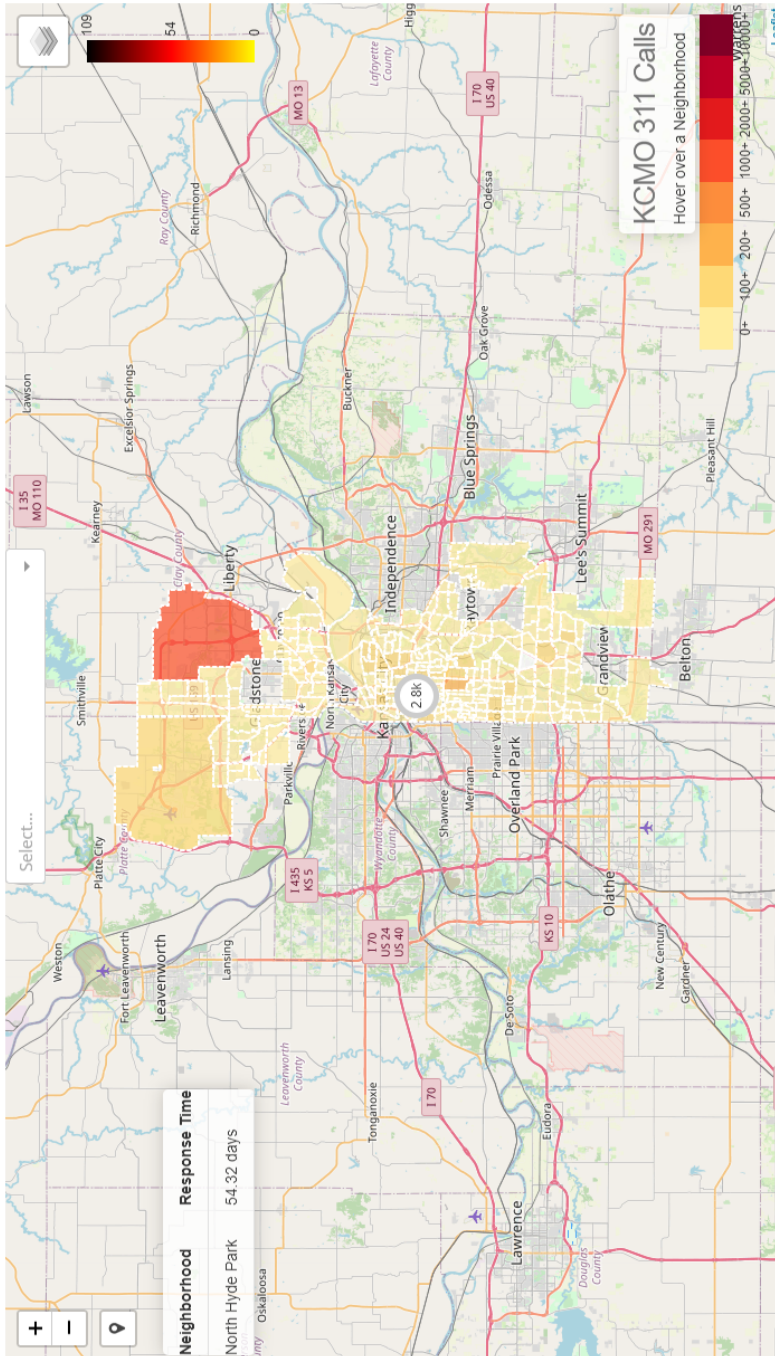


Figure 28: Map Visualization

### 4.1.3 Temporal Visualization

The color of each block represents the call volume at that particular geographic entity (a neighborhood in this case), starting from a lighter shade to a darker shade. It acts as a first-level indication of the performance of the neighborhood. The color of the neighborhood invites the user's attention naturally to the neighborhood with the highest call volume, which correlates to a higher number of issues. To further explore, one can use the graphs to identify trends at departmental, categorical, and the total volume level, provided with the help of bar charts and line charts.

Using order preserving hierarchical clustering Algorithm, each 311 call's location is used to cluster the calls at a location based on the zoom level. There are 19 levels the user can access, starting at 0 (for the whole world), ending at 18 (for the most granular tile). Based on the geographic entity's area under consideration, the number of clusters provided by the partition algorithm differs based on the zoom level. For a small area number of partitions get clustered faster than they do for a larger area. The clusters are built bottom-up by considering the Euclidean distance of positional data from a lower level and identifying points which are medians of the clusters in the layer below, equidistant from the median points.

The overall tool acts as a one-stop go-to point (dashboard) for analyzing the 311 call status in a city at a level of neighborhoods, block-groups, and counties. At a high level, it provides quantitative analysis of the requests and complaints at a neighborhood level from the updated publicly available data and a picture of the performance of governance on different parameters (call volume, categories, departments, etc.) and at the



granular level, it acts as a tool to drill down and understand what the reason behind the picture that is visible as such, by exploring the 311 calls at an individual call level. Each component has its own story to tell the user about its insights. Combining such elements provides a comprehensive understanding of the current standing of the neighborhood's state-backed and the reasons behind the same by mapping them with 311 calls reported in the database.

The spatial partitioning algorithm abstracts away the fine-grain details and helps gain a higher level of insight into the geographical distribution of 311 calls. The 311 call data considered contains information about many categories, which have been cleaned and mapped to 246 in the final version. A set of 16 to 64 nodes are clustered together at the base level, compounding the number of nodes at each level:  $16 \rightarrow 16^2 \rightarrow 16^4 \rightarrow 16^8$  and so on. At a higher level, dealing with individual calls would require a lot of computing resources. This approach normalizes the number of data points at each level, making it easy to apply the computing power requirement.

#### 4.1.4 Trend Visualization

**Call Volume Trend:** Figure 29 shows the call volume trend, which plots the number of calls received over the chosen period in the filters. Call volume can be a direct indicator of the neighborhood activity.

**Yearly Composition of 311 Call Types:** Figure 30 shows the composition of types of 311 calls in the time period from 2007 to 2020 divided into three bins - 1) 2007 - 2010 (Green), 2) 2011 - 2015 (Orange) and 3) 2016 - 2020 (Purple). Each bin corresponds to

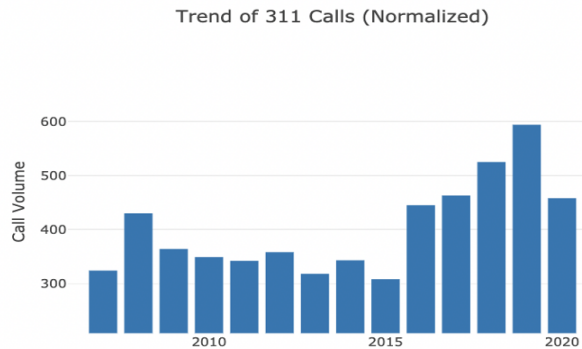


Figure 29: 311 Call Volume Trends for Years from 2007 to 2020

the percentage of calls for a type of 311 calls by volume in total calls received. It provides insights into how the composition has evolved through the years. If a type of call had more incidents in the past bin than the recent bin, it might signify that the problem of that type has been taken care of, or there is a measure put in place to reduce the effect of the issue. If it is the other way around, i.e., if the percent of call volume is higher in the recent bin than the past years' bin, it signifies that it is a rather recent development and a need to address the issue to stop it from growing further.

**Distribution of 311 Call Timings:** Often, it may be helpful to identify the busy hours of any business. Then, why not for public data? Especially, to scale the existing solution, it helps identify and increase maximum utilization to provide a better quality of service to the public. To accommodate this idea, we have visualized the call traffic on a clock, using a radar chart in Figure 31. This chart is also responsive and gets updated when the filters for time and neighborhood are changed. The peaks in this graph are when there is a large amount of activity for 311 calls service. Strategically increasing the resources allocated during the peak hours would help serve more public and improve the satisfaction index.

### Top Request Types - Composition

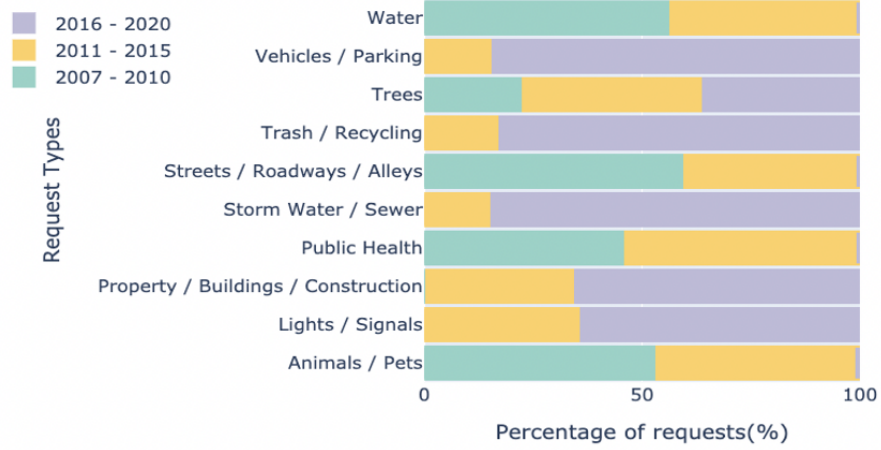


Figure 30: Yearly Composition of 311 Call Types

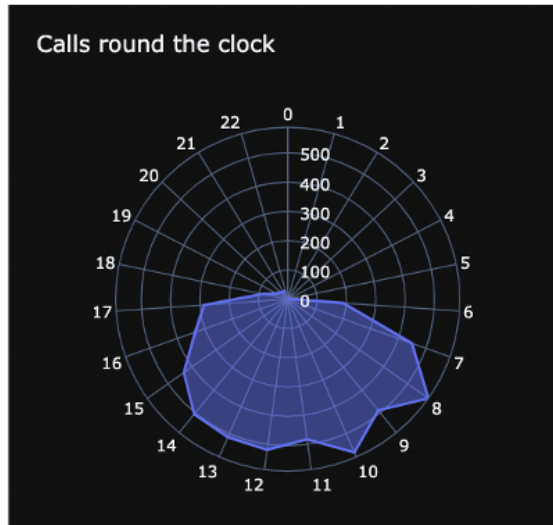


Figure 31: 311 Calls - Round the Clock.

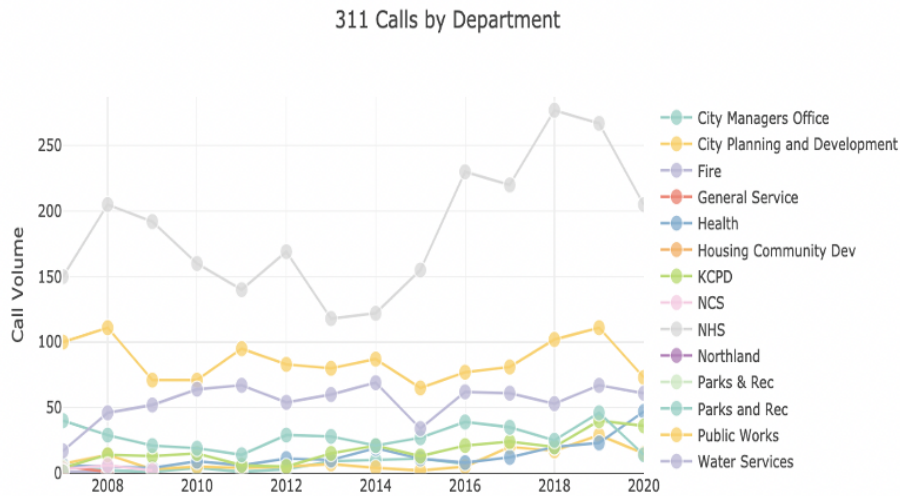


Figure 32: Departmental Call Volume Trend

**Departmental Call Volume Trend:** As a part of the resolution process, once a 311 call is received, it is routed to the concerned department. The department then takes over and handles the concern, provides a solution, and closes the issue. The number of calls dealt with by each department offers insights into the workload, demand for resources, and need for collaboration which are essential metrics for good governance. Therefore, the call volume trends for different departments may be thought of as indicators of the respective department’s performance. A visual comparison of call volume based on departments is presented in Figure 32.

**Categorical Call Volume Trend:** Each call is assigned a category similar to how it is given a department. The departments define these categories to identify, label quickly, and organize the types of 311 calls, which helps streamline and promptly resolve issues. These categories correspond to the type of the case. Analyzing the call volume based on variety provides insights into the type and intensity of the problems that prevail in a

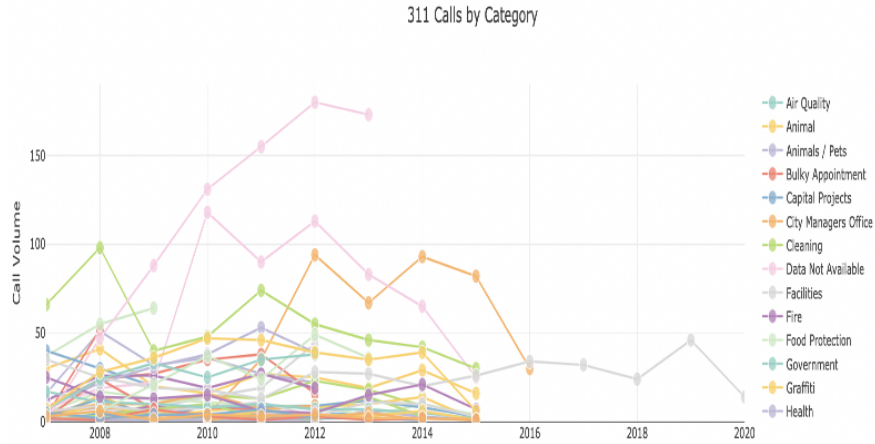


Figure 33: Categorical Call Volume Trend

neighborhood. The more the problems, the higher the need to address and focus on the issue. A sample of the definite call volume trend for all neighborhoods can be observed in Figure 33.

#### 4.1.5 Analysis of the 311 calls data:

One of the most used data visualization approaches is using Tableau to generate graphs with the available data. But the functions and operations in Tableau are limited in many ways that are not as flexible as the scripts written from scratch. Since the data in our case exhibited several challenges, we figured we needed a framework with more control over the data while producing a high-quality interactive representation of insights. So, we decided to evaluate Dash (by Plotly) for its data visualization capabilities and full-stack product development. Dash provides a highly integrated framework focused on data visualization with scalable front-end and back-end capabilities. It uses React JS to

provide a component-based development framework for front-end and Flask - a Python-based framework, easy to set up and extend for back-end with tight integration between the two. As a result, it helps streamline the development of the application with fast development cycles.

**Map Component:** The leaflet is an open-source map provider, supporting both ReactJS and Dash. It has been widely adapted for its extensibility and active community support. The neighborhood data is provided to the Leaflet Map Component as GeoJson, an industry standard for representing geographical mapping and boundary information. The insights derived from the raw data are converted into a compatible format before mapping. It is then provided to the user to choose from World Imagery, Gray Canvas, Topology, and Terrain base maps. Events generated by choosing (by clicking) and hovering on an active item (polygon, boundary, layers, etc.) are used to update the data across all the application charts.

**Chart Components:** Dash provides components for Bar Chart (used for Volume Trend), Radar Chart(used for Round the Clock Trend), Line Chart (used for Department and Category Trend), and Horizontal Bar Chart (used for Composition Trend) to work with different kinds of data and fit a wide range of visual insights. Thanks to Flask's lightweight and fast server, the application comes to life in a snap when combined with the AWS cloud's power.

**NLP Components:** Topic Modelling, LDA, and Bigram analysis take a reasonable amount of time. Hence they are precomputed for the whole data, and those results are accessed only when needed. It saves tremendous time and resources. The results are published on

an interactive chart and data-table with integration across other components (timeline and drop-downs).

#### 4.1.6 Training:

To train the 311 BERT classification models of categories and departments of the 311 service requests, the 311 data was split between training and validation in the ratio of 80-20. As a result, the 311 BERT classification model has been trained in 10 epochs. The accuracy of the 311 service category and department classification was approximately 95.5% and 96.15%.

Example schema of 311 calls:

```
{
  ``case_id": ``2021033686",
  ``source": ``PHONE",
  ``department": ``NHS",
  ``work_group": ``NHS-Dangerous Buildings-",
  ``request_type": ``Prop/Build/Construct-Dangerous Building-On list",
  ``category": ``Property / Buildings / Construction",
  ``type": ``Dangerous Building",
  ``detail": ``Standard",
  ``creation_date": ``2021-10-28T00:00:00.000",
  ``creation_time": ``10:40 AM",
  ``creation_month": ``10",
  ``creation_year": ``2021",
  ``status": ``OPEN",
  ``exceeded_est_timeframe": ``N",
  ``street_address": ``2639 Garfield Ave",
  ``address_with_geocode": {
    ``latitude": ``39.077701",
    ``longitude": ``-94.557926",
    ``human_address": ``{"address": \"2639 Garfield Ave\",
    \"city\": \"\", \"state\": \"\", \"zip\": \"64127\"}"
  },
}
```

```

    ``zip_code": ``64127",
    ``neighborhood": ``Wendell Phillips",
    ``county": ``Jackson",
    ``council_district": ``3",
    ``police_district": ``Central",
    ``parcel_id_no": ``14494",
    ``ycoordinate": ``39.077701",
    ``xcoordinate": ``-94.557926",
    ``case_url": {
      ``url": ``http://city.kcmo.org/kc/ActionCenterRequest/CaseInfo.aspx
      ?CaseID=2021033686"
    },
    ``days_open": ``0",
    ``:@computed_region_kk66_ngf4": ``234",
    ``:@computed_region_9t2m_phkm": ``3",
    ``:@computed_region_my34_vmp8": ``7",
    ``:@computed_region_w4hf_t6bp": ``95"
  },

```

This data is further analyzed according to the locations and time of the 311 service requests. The 311 service requests have been projected over the map according to the ZIP code of the requests. Figure 28 shows the frequency of the claims according to the color intensity (the higher frequency, the darker red). The 311 service requests have been summarized over the years for each department, as shown in Figure 27.

For example, consider the following sentence (1) *A Computer Science Master's student ("I")*; (2) *AI: The machine learning models built using the data relevant to the life story*; (3) *Residents in KC who are looking for a solution for their problem*; (4) *City department who can provide solutions to Residents*; (5) *Neighborhoods who are living near to the residents*.

- **What happened:** predictive models to identify the city department who can handle



their problems.

- **When:** Time Series models to estimate the time to respond to it. The models will be built using the data that are used considering the time.
- **Who & Where:** the skills and expertise of community members; streets, wifi, pipes, etc.; churches, non-profits, HOAs, etc.; businesses, etc.
- **Why & How:** predictive models for possible causes and consequences.

All the charts and maps are connected with a temporal filter, allowing the time-based analysis of 311 data at both yearly and monthly granularity levels. Once chosen, the year and month filter gets applied to the call volume, departmental and categorical trends. The filtered data is available to download for further analysis at the click of a button. The first level of analysis of call volume follows a logical analysis of departments and category-based research. Call volume by departments shows the performance of departments in handling the issues around the neighborhoods. Call volume by category offers the performance of departments in addressing the problems around the neighborhoods.

#### 4.1.7 Predictive Models for What Happened

The first step to problem-solving is to identify the problem: What happened? The 311 Calls in Kansas City (called *311KC*) are communities' input about their neighborhood problems. Our goal is to build predictive models to answer two critical questions for community members who sent in the requests- *Who can handle a particular 311 call? How long will be taken to address the issue?* We will present (1) how the 311call data

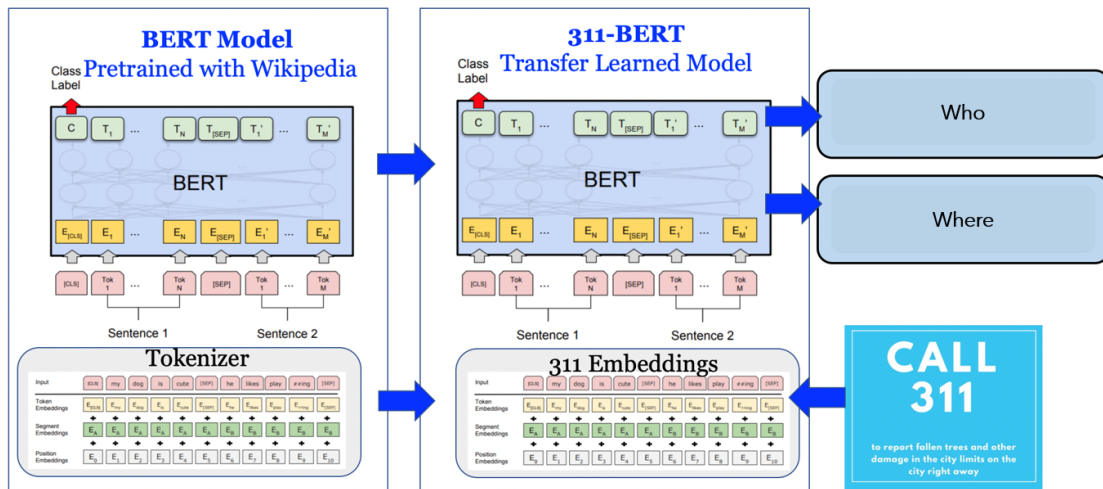


Figure 34: Architecture of Who & Where BERT Auto Annotation Model

can be converted into *311KC* story that is composed of a set of questions and (2) how to build predictive models to answer two critical questions - *Who can handle a particular 311call? How long will be taken to address the issue?*

311 is a service provided by the government of the USA for dealing with nuisance updates. Individuals can contact the 311 services by Phone, Online website, or from Twitter to report any nuisance cases in their neighborhood. These cases are updated daily on Open Data KC - a repository of publicly available data. Thus, we created the BERT-311 Model (Bidirectional Encoder Representations from Transformers) [24]. BERT has received research attention due to its state-of-the-art performances in a wide variety of NLP tasks [24]. The purpose of the model is to identify the current key problems people are facing in their daily life. Thus, predictive deep learning models need to understand the context of the problems (impacts, the complexity of problems, associated incidences, etc.). The BERT-311 model includes two classification models of a 311 call description,

its category of 311 services, and the department of the service request.

Two models were designed to determine what happened and predict the nature of the problem and the working groups of the city agencies to handle the situation: 1) A predictive model of the functional groups of the city agencies, and 2) a predictive model of the type of problems based upon the 311 call taxonomy. The two models were designed using BERT (Bidirectional Encoder Representations from Transformers) [24] that has received research attention due to their state-of-the-art performances in a wide variety of NLP tasks.

Our proposed architecture (shown in Figure 34) is based on the combination of the BERT models, which is the bidirectional training of Transformer that focuses on building a useful attention model in language modeling, and the domain-specific embedding. The BERT's bi-directional attention model is different from the existing training in text sequence, either left-to-right or right-to-left. The language the model trained bidirectionally is superior to language context in a more profound sense than single-direction language models. Our model was designed with bidirectional training in models with the domain-specific embedding.

Formally, for single sentence classification and tagging tasks, the segment embedding has no discrimination. A particular classification embedding ([CLS]) is inserted as the first token, and a unique token ([SEP]) is added as the final token. A 311 service description is inserted as the first token, and the 311 call domain-specific embedding is added as the final token. Given an input token sequence  $x = (x_1, \dots, x_T)$ , the output of BERT is  $H = (h_1, \dots, h_T)$ . Our classification model was designed with BERT with joint

classification and slot filling.

Based on the hidden state of the first special token ([CLS]), denoted  $h_1$ , the intent is predicted as:  $y_i = \text{softmax}(W_i h_1 + b_i)$  For slot filling, we feed the final hidden states of other tokens  $h_2, \dots, h_T$  into a softmax layer to classify over the slot filling labels. To make this procedure compatible with the WordPiece tokenization, we feed each tokenized input word into a WordPiece tokenizer and use the hidden state corresponding to the first sub-token as input to the Using BERT for a specific task is relatively straightforward: BERT can be used for a wide variety of language tasks while only adding a small layer to the core model:

In the training process, the pairs of sentences as input are processed and learns to predict if the second sentence in the pair is the subsequent sentence in the original document. During training, 50% of the inputs are a pair in which the second sentence is the following sentence in the original document, while in the other 50% a random sentence from the corpus is chosen as the second sentence. The assumption is that the random sentence will be disconnected from the first sentence. To help the model distinguish between the two sentences in training, the input is processed in the following way before entering the model:

A [CLS] token is inserted at the beginning of the first sentence, and a [SEP] token is inserted at the end of each sentence. A sentence embedding indicating Sentence A or Sentence B is added to each token. Sentence embeddings are similar in concept to token embeddings with a vocabulary of 2. A positional embedding is added to each token to indicate its position in the sequence. The idea and implementation of positional

embedding are presented in the Transformer research [56].

Classification tasks are done similarly to Next Sentence classification by adding a classification layer on top of the Transformer output for the token. In Question Answering tasks (e.g., SQuAD v1.1), the software receives a question regarding a text sequence and is required to mark the answer in the series. Using BERT, a Q&A model can be trained by learning two different vectors that mark the beginning and the end of the solution. In Named Entity Recognition (NER), the software receives a text sequence and is required to mark the various types of entities (Person, Organization, Date, etc) that appear in the text. Using BERT, a NER model can be trained by feeding each token's output vector into a classification layer that predicts the NER label.

**Why** - The problem will be further analyzed before and after the instances and then annotated as (**Causes** and **Consequences**). Causes can be defined as circumstances or situations that cause the sample to happen. Consequent indicates the impact on future instances. Causes are immediate if the events are most closely connected with the instance, while the causes are underlying if it is less directly contributed to the instance.

For local governments to learn about their cities from 311 data, it is going to necessitate examining the spatial distribution of aggregated contracting volume. Although 311 data are deficient in several ways, the openness of the program and the geographical precision of the data offer numerous possibilities for understanding government action and city life. Prior research had identified individual-level causes of citizen-government contacting, but the extent to which these propensities are connected to actual geographic distributions had not been considered. Indeed, the results presented above are strongly

suggestive of patterns in the distribution of 311 contacting volume and should inform how we think about both local civic engagement and the condition of government-provided goods in neighborhoods. This analysis may also help set the stage for other studies of citizen contacting that seek to understand its causes and what it tells us about the spaces we live in.

“if contacting occurs in response to the perception of unsatisfactory conditions, then perceived unsatisfactory conditions need to be present to induce the contacting. Whether it is potholes, dirty sidewalks, downed trees, graffiti, noise, or any other problem, people need a reason to contact the city before they contact the city. Causes connected with the attitudes of the people who live in a space primarily include what can be broadly defined as “contacting propensity. At the individual level, contacting propensity is the likelihood that a person will contact the government using 311 independent of the actual conditions. The question of “where are conditions poor?” and the question of “what people contact the government?” cannot be entirely separated. Perhaps more frustratingly, Can we answer whether a social or demographic characteristic (e.g., income, race) of a space is a cause of increased/= or decreased propensity or a cause of better or worse conditions. Although the causal direction is an issue, the analysis will be able to show the extent to which attributes explain the geography of 311 contacting and are consistent with different theoretical expectations associated with both conditions and propensity.”

**How** - how to handle the instances and defend them for any future incidences For the problem story, we need data to describe and understand the problems in depth.

Data searching and sharing relate to important aspects that determine where the

<b>Who</b>	Which department or which team will be in charge of the problem? A predictive model will be built to determine who can handle the situation.
<b>When</b>	How long does it take to handle the problem? A predictive model will be built to determine the response time.
<b>Where</b>	Where should it be resolved first?
<b>Why</b>	Why do we have to handle it? A predictive model will be built to determine the consequence of the problem if this does not take it right now.
<b>How</b>	How to solve the problem?

Table 5: 5W1H Components Identification - 311

data would come from (searching for publicly available data or sharing owned/private data), what kind of data would be used, how much data would be required, what format of data should be prepared, and how they could be characterized.

#### **4.1.7.1 Objective 3: StoryNet Application.**

The two predictive models mainly focus on the potential solutions for the problems, meaning based upon “what happened.” defined by PS in Objective 2 and to estimate the cost or times required for the solution, further to prioritize the solutions based on the contexts. For the Solution Story, we need data to describe and understand the problems in depth.

Data searching and sharing relate to important aspects that determine where the data would come from (searching for publicly available data or sharing owned/private information), what kind of data would be used, how much data would be required, what format of data should be prepared, and how they could be characterized. The predictions from classification model for department and problem categories are used to identify the

corresponding model for prediction with the Prophet forecasting model [98], which is a nonlinear statistical regression based upon time series analysis tool. The Prophet model was built to predict the estimated responding time for a specific 311 service call using the 311 service data from the past ten years.

The Prophet model has major components, including Growth forecasting that is a model to understand how the population has grown and will be continuously growing. This is modeled using the piece-wise logistic growth model as follows: The trend changes in the growth model have been adjusted by explicitly defining change points where the growth rate is allowed to change. For given  $S$  change points at times  $s_j, j = 1, \dots, S$ , a vector of rate adjustments is defined as  $\delta \in R^S$ , where  $\delta_j$  is the change in rate that occurs at time  $s_j$ . The rate at any time  $t$  adjustments up to that point:  $k + P_j : t > s_j \delta_j$ . A vector is defined as  $a(t) \in \{0, 1\}^S$  and the rate at time  $t$  is then  $k + a(t)^T \delta$ . When the rate  $k$  is adjusted, the offset parameter  $m$  is also adjusted to connect the endpoints of the segments. The correct adjustment  $\gamma$  at change point  $j$  is defined as follows:

$$g(t) = \frac{C(t)}{1 + \exp((k + a(t)^T \delta)(t - (m + a(t)^T \gamma)))} \quad (4.1)$$

with  $C(t)$  is time-varying capacity,  $k$  the growth rate, and  $m$  an offset parameter. Also, in order to fit and forecast these effects, seasonality models have been defined as periodic functions of  $t$  using Fourier series considering periodic effects.

This model was built to identify the time series patterns in 311 service resolution time based on the times (daily, weekly, and yearly seasonality) for the service category and service department from the 311 predictive models. We have also explored several



deep learning models, including LSTM and CNN models, for the time series forecasting. Surprisingly, the Prophet forecasting model has shown superior prediction compared to the deep learning models (Table 9).

#### 4.1.8 Identifying “What”

We have proposed a topic prediction model by building the contextual topic approach based on the integration of LDA [10] and BERT [24]. This model aims to identify the topic that the model would develop and plot a corresponding word cloud for each of the  $K$  unique categories where the number of issues  $K$  was determined by the optimal topic model based on KL-divergence. The dominant topic for each description is identified by matching the probability of each topic with the description. These plots can be mapped to the 311 categories in our domain.

Latent Dirichlet Allocation (LDA) [10] is a generative probabilistic model based on the three primary structures, including word, topic, and document. A distribution over topics is determined with the documents while a distribution over words with each topic. To generate a document, LDA firstly samples a document-specific multinomial distribution over topics from a Dirichlet distribution. Then it repeatedly samples the words from these topics. LDA and its variants have been successfully applied in many works [2], [10], [15], [16].

Given an input corpus  $D$  with  $V$  unique words and  $M$  documents, each document containing a sequence of words  $dw_1, w_2, \dots, w_{Nd}$ . Given an appropriate topic number  $K$ , the generative process for a document  $d$  is as following: Sample a K-vector

$\delta_d$  from the Dirichlet distribution  $p(\delta|\alpha)$ , where  $\delta_d$  is the topic mixture proportion of document  $d$ . For  $i = 1 \dots N_d$ , sample word  $w_i$  in the document  $d$  from the document-special multinomial distribution  $p(w_n|\delta_d, \beta)$ , where  $\alpha$  is a  $k$ -vector of Dirichlet parameters, and the Dirichlet distribution  $p(\delta|\alpha)$   $\beta$  is a  $K \times V$  matrix of word probabilities, where  $\beta_{ij} = p(w_{j=1}|z_{i=1}), i = 0, 1, \dots, K; j = 0, 1, \dots, V$ .

In LDA, the topic proportions are randomly drawn from a Dirichlet distribution, which implies the independence between topics. However, these correlations are widespread in real-world data. Interestingly, the association or correlation of topics can be modeled with LDA. The topic “311 services” is often correlated with “crimes” while unlikely co-occurs with “business.” There would be an inconsistency between the assumption and input documents so that the predefined parameter  $K$  may not be able to reflect the real topics of the domain. To overcome the limitation, the NLP embeddings, such as BERT, could be significantly contributed to determining relevant or non-related topics in the 311 call domain. CTM replaces the Dirichlet distribution with Logistic Normal one. After getting the correlation between every pair of topics through the covariance matrix, CTM can predict not only the words generated by the same topic but also the words caused by the correlated issues. Compared with LDA, CTM is less sensitive to  $K$ , but both cannot automatically select the number of topics.

StoryNet is an effective instrument for co-creation and co-teaching in STEM education. The StoryNet is designed as a knowledge graph so that the underlying representations of the stories are W3C standard Resource Description Framework (RDF) and RDF

Schema [57]. The similarities of stories are determined by NLP and topic modeling techniques based on their common contexts. The 5W+H questions will be mapped to NLP-based question, and the answer embedding and the story embedding will be mapped to SPARQL and Shapes Constraint Language (SHACL), which allows for a semantic query for retrieving and manipulating data stored in Resource Description Framework format.

For the 311 Q/A model, we used ALBERT [47] in which the backbone is similar to the transformer encoder [100] with GELU nonlinearities [38] built in BERT. We follow the BERT notation conventions and denote the vocabulary embedding size as  $E$ , the number of encoder layers as  $L$ , and the hidden size as  $H$ . Following Devlin et al. (2019), we set the feed-forward/filter size to be  $4H$  and the number of attention heads to be  $H/64$ . Our innovation is to use the predictive models such as BERT-311, the time series forecasting model, and the topic identification model to build the AI-Community 311 Q&A framework based on the ALBERT question answering. Our design of the Q&A follows the story framework presented earlier.

We applied story framework to the design thinking of answering each incoming 311 call. Accordingly, we created a question answering system that can 1) conduct interactive sessions of Q&A, 2) answer some open-ended questions like *What happened?*, *When is it happened?*, *Who can handle it?* *Where does it happen?* *How long does it take to work?* *Could you tell me more about it?*. based on the AI models generated by the OCEL.AI story framework, 3) an AI collaborator, which uses Q&A to assist and guide 311 agencies to response to calls. In addition to the Q&A features, the interactive visualization dashboard will have great potential for visual learning and understanding the

content of the story framework.

Figure 35(a) shows the connection of 311KC to other stories, such as “Used Cars” and “Roads in Michigan,” which are also a part of the OCEL AI project. The graphs are generated by identifying each story’s most representative terms using LDA (Latent Dirichlet Allocation). The stories are connected through the common topics to help identify new insights, which may help create a new story. Figure 35(b) shows the data sources that were used for building models and apps for various stories. Figure 35(c) shows the model network, which identifies the relations between different stories through the lens of machine learning models and techniques used in them.

Such a network helps to identify the most suited models for a similar scenario based upon the use case. The standard methods used among different use cases help students select machine learning solutions intuitively based on the real-world impact rather than just learning the syntax. Figure 35(d) shows the relations among the applications of stories through the tools used to implement them. This graph also represents solutions to real-world problems.

From this graph, the students can identify relevant and useful applications for each use case. Figure 35(e) shows the network of concerns and causes related to the ethical considerations for each use case. These graphs add a layer of social and cultural depth to use cases and display a comprehensive picture of the solution in a broader social context. Figure 35(f) shows the five stories together as OCEL.AI StoryNetwork.

Latent Dirichlet Allocation (LDA) [10] is one of the most popular approaches.





neural networks is cross-entropy to measure the difference between the predicted labels and the proper labels; (ii) Accuracy Eq 4.3.

$$Cross - entropy = - \sum_{i=1}^m \sum_{j=1}^n y_{i,j} \log(p_{i,j}) \quad (4.2)$$

where  $y_{i,j}$  denotes the true value  $p_{i,j}$  denotes the probability predicted by the model of sample  $i$  belonging to class  $j$ ,  $m$  is the number of the classes, and  $n$  is the size of a training set.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (4.3)$$

where TP is true positive, FN is false negative, FP is false positive, and TN is true negative.

GLUE Benchmark includes 9 natural language understanding tasks:

### **Single-Sentence Tasks:**

CoLA - The Corpus of Linguistic Acceptability [108] is a set of English sentences from published linguistics literature. The task is to predict whether a given sentence is grammatically correct or not.

SST-2 - The Stanford Sentiment Treebank consists of sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence: positive or negative.

### **Similarity and Paraphrase tasks**

MRPC - The Microsoft Research Paraphrase Corpus [29] is a corpus of sentence pairs automatically extracted from online news sources, with human annotations for whether

the sentences in the pair are semantically equivalent.

**QQP** - The Quora Question Pairs [19] dataset is a collection of question pairs from the community question-answering website Quora. The task is to determine whether a pair of questions are semantically equivalent.

**STS-B** - The Semantic Textual Similarity Benchmark [114] is a collection of sentence pairs drawn from news headlines, video, and image captions, and natural language inference data. The task is to determine how similar two sentences are.

### **Inference Tasks**

- These are the inference tasks identified for analysis of the models:

**MNLI** - The Multi-Genre Natural Language Inference Corpus [63] is a crowd-sourced collection of sentence pairs with textual entailment annotations. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). The task has the matched (in-domain) and mismatched (cross-domain) sections.

**QNLI** - The Stanford Question Answering Dataset [81] is a question-answering dataset consisting of question-paragraph pairs, where one of the sentences in the paragraph (drawn from Wikipedia) contains the answer to the corresponding question. The task is to determine whether the context sentence contains the answer to the question.



**RTE** - The Recognizing Textual Entailment (RTE) datasets come from a series of annual textual entailment challenges. The task is to determine whether the second sentence is the entailment of the first one or not.

**WNLI** - The Winograd Schema Challenge is a reading comprehension task in which a system must read a sentence with a pronoun and select the referent of that pronoun from a list of choices (Hector Levesque, Ernest Davis, and Leora Morgenstern. The winograd schema challenge. In Thirteenth International Conference on the Principles of Knowledge Representation and Reasoning. 2012).

All tasks are classification tasks, except for the STS-B task which is a regression task. All classification tasks are 2-class problems, except for the MNLI task which has 3-classes.

The evaluation of the proposed predictive models has been conducted in comparison to other machine learning algorithms, including Support Vector Machines (SVM), Decision Tree, Naive Bayes, K-means Clustering.

The confusion matrix for these two models is shown in Figure 39. The class-wise accuracy for the 311 service category and department classification are shown in Figures 37 and 38. The Recreation and Park category has the lowest accuracy (82.61%) in the 311 category classification. The north department is the least accurate department (67.50%) among the departments.

Table 6 shows the performance comparison on GLUE benchmark with different BERT approaches. Score (ensemble) is calculated as the average of CoLA, SST, MRPC, STS, QQP, MNLI, RTE, and WNLI metrics.

Model	Train FLOPs	Score
BERT	1.9e20 (0.06x)	79.8
RoBERTa	3.2e21 (1.02x)	88.1
ALBERT	3.1e22 (10x)	89
XLNet	3.9e21 (1.26x)	89.1
AllenNLP	3.88e21 (1.25x)	87.3
StoryNet (Ours)	3.1e21 (1x)	89.5

Table 6: Performance Comparison - GLUE Test

Task	Metric	ALBERT	RoBERTa	StoryNet	BERT base	BERT Ig
CoLA	Matthew's correlation	54.94	61.72	64.56	52.1	60.5
SST-2	Accuracy	92.74	91.86	95.87	93.5	94.9
MRPC	F1/Accuracy	92.05/88.97	91.87/88.61	92.36/89.46	88.9/-	89.3/-
STS-B	Pearson/Spearman corr.	90.41/90.21	90.07/90.10	91.51/91.61	-/85.8	-/86.5
QQP	F1/Accuracy	88.26/91.26	88.80/91.65	89.18/91.91	71.2/-	72.1/-
MNLI	Matched/Mismatched	86.69/86.81	88.66/88.73	89.86/89.81	84.6/83.4	86.7/85.9
QNLI	Accuracy	92.68	93.66	94.33	90.5	92.7
RTE	Accuracy	80.87	82.86	83.39	66.4	70.1

Table 7: Score Breakdown

The scores presented here are:

**Matthews Correlation Coefficient (MCC):** The Matthews Correlation Coefficient is used in machine learning as a measure of the quality of binary (two-class) classifications. It takes into account true and false positives and negatives and is generally regarded as a balanced measure which can be used even if the classes are of very different sizes. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between -1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and -1 indicates disagreement between prediction and observation.

The formula for the Matthews Correlation Coefficient is:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \quad (4.4)$$

where, MCC is the Matthews Correlation Coefficient and

TP	True Positives
FP	False Positives
TN	True Negatives
FN	False Negatives

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.5)$$

$$Precision = \frac{TP}{TP + FP} \quad (4.6)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.7)$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} = \frac{2 * TP}{2 * TP + FP + FN} \quad (4.8)$$

**Spearman correlation:**

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4.9)$$

**Spearman’s correlation** assesses the degree of linear association between two variables measured at the ordinal level. More specifically, it is assess the linear association of the ranks for a paired sample  $(X_n, Y_n)$

The formula for Spearman’s correlation is given as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \quad (4.10)$$

where  $d_i$  = difference in paired ranks and  $n$  = number of cases

**311 Call Classification Results**

The class-wise accuracy for the 311 service category and department classification are shown in Figures 37 and 38. The Recreation and Park category has the lowest accuracy (82.61%) in the 311 category classification. The north department is the least accurate department (67.50%) among the departments. As shown in Table 8, the BERT classification models performed the best. SVM was the second-best among the five different algorithms (92.96% and 93.77%), and Decision Tree was the worst (44.66% and 62.95%).

**Discussion about 311 Call Classification Models** Figure 39(a) shows that categories with lower accuracy (e.g., “parks & recreation”) were largely due to data imbalance issues. Categories with higher frequencies tended to have the highest accuracy, while those with lower frequencies had the lowest accuracy. Although “Property Violations” enjoyed a 99.7% accuracy, the miscategorized cases suggested some overlapping with “Street”

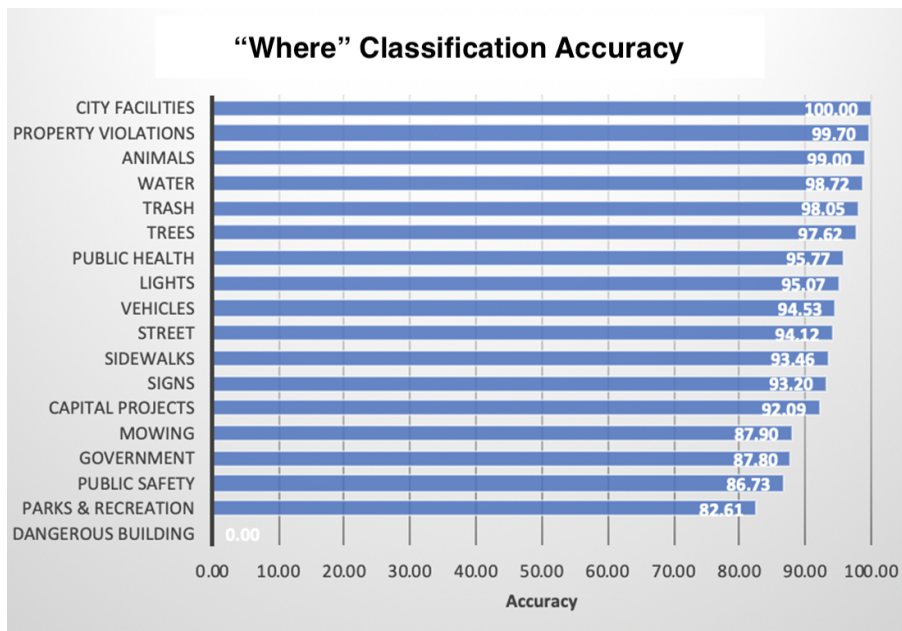


Figure 37: “Where” Class-Wide Classification Accuracy

and “Trash.”

Figure 39(b) shows departments with higher frequencies had higher prediction accuracy. Among the lowest four departments, “Finance,” and “South” had fewer than 60 cases. The confusion occurred mostly between “Northland” and “City Managers Office” and “NHS (National Honor Society).” The mixed-up can be attributed to the crossover between function-based department classification (like National Honor Society) and jurisdiction-based department classification.

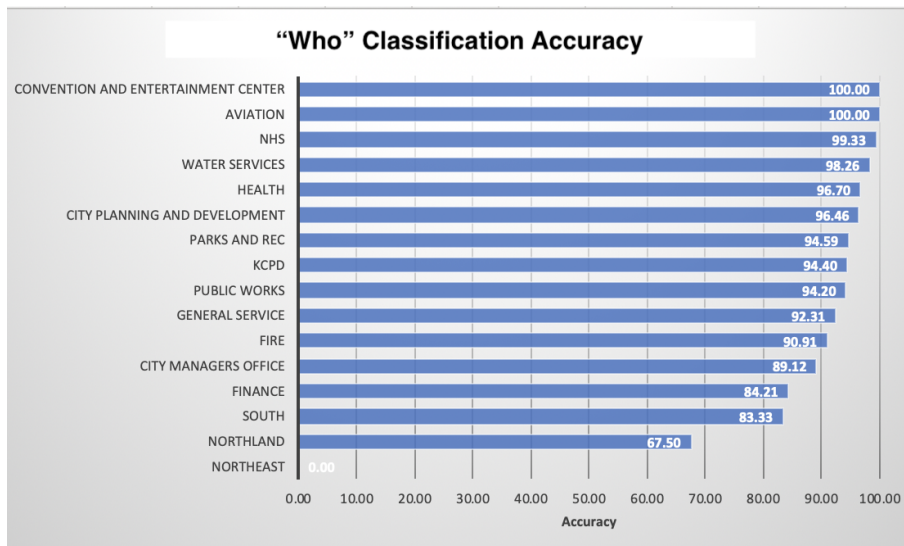


Figure 38: "Who" Class-Wide Classification Accuracy

Table 8: 311 Who and Where Prediction Accuracy

Model Name	Who	Where
Naive Bayes	81.46%	85.89%
K-means Clustering	82.91%	85.08%
SVM	92.96%	93.77%
Decision Tree	44.66%	62.95%
BERT Transformer	<b>95.50%</b>	<b>96.15%</b>

#### 4.1.10 "When" Model: KCMO 311 Time Series Forecasting

The predictions from the classification model for department and problem category are used to identify the corresponding model for prediction with Prophet - a statistical nonlinear regression-based time series analysis tool. The Prophet model identified is used to predict the estimated time to resolution. It has been trained on the data from the past ten years to identify the patterns in resolution time based on daily, weekly and yearly seasonality.

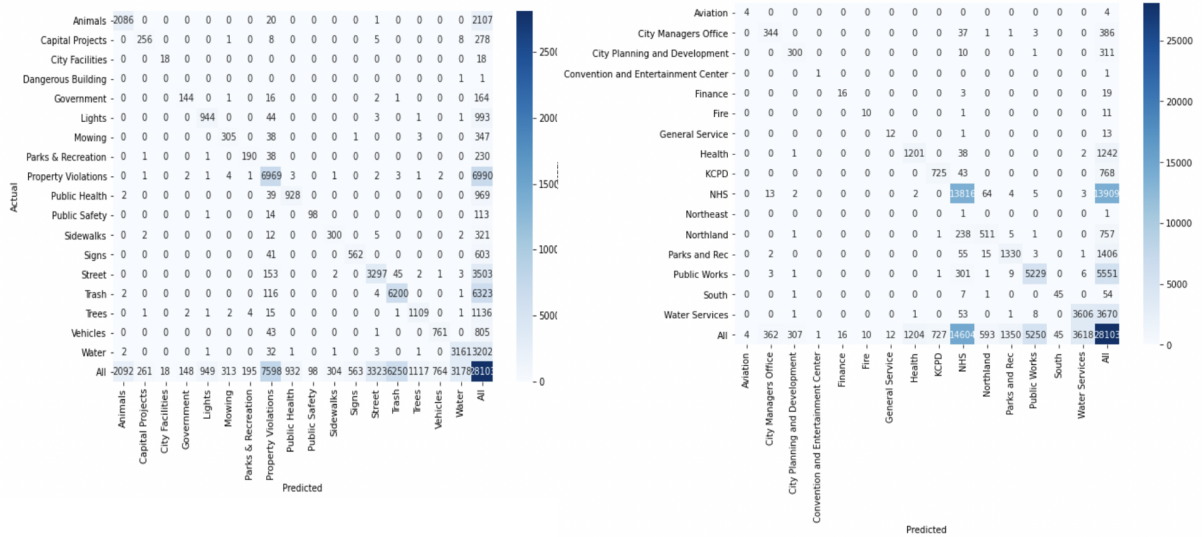


Figure 39: Confusion Matrix: (a) Where (b) Who

**Data:** The time series forecasting was conducted on all the data available from 2015 to 2020. The results are shown in Figure 18. It shows a steadily decreasing trend overtimes, which suggests an improvement in the performance of the city service. The cases that were used to take around 80 days to resolve in 2015 were solved in under 20 hours in 2020.

**Training:** To further drill down upon the trends of the case resolution, we have analyzed the same data based upon the category of the 311 service requests. Since we do not have continuous data from the whole date range, due to lack of data, only cases since 2008 were entered into the analysis. Therefore, the category-wise trends have been shown in Figure 40.

This figure shows that the most significant improvement were Animals, Government, Mowing, Vehicles, and Water. Capital Projects, Public Safety, Sidewalks, Signs, and



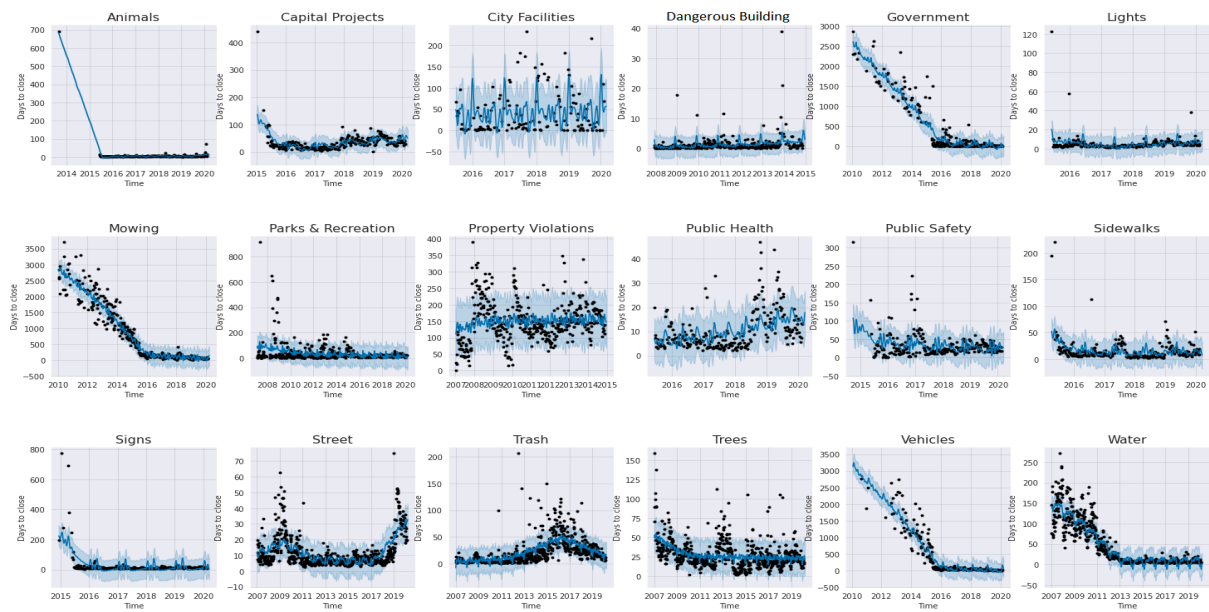


Figure 40: 311 Service Category Wise Time Prediction

Trees have a slower and Lights exhibit strong seasonality, without a significant downward or and Trash show highly idiosyncratic patterns; the outbreaks of the problems rather than seasonality influences service time. These are most problematic areas of 311 service delivery. Figure 37 and Figure 38 shows accuracy for the training and validation for both Department Classification and Problem Classification.

In conjunction with the overall trend, many of the categories exhibit the steady decrease of the resolution time (in days) over time from the beginning of the data availability, with *Public Health* being an exception. The issues related to *Trash* have been observed to rise during four years of 2014-2018, but have subsided after that. The issues related to *Lights*, *Animals*, *Capital Projects*, *Sidewalks*, *Trees*, *Animals* and *Property Violations* have been showing a steady trend. The steep decreasing trend for *Vehicles*, *Mowing*, *Government*, and *Water* issues shows that there have been very effective responses to the

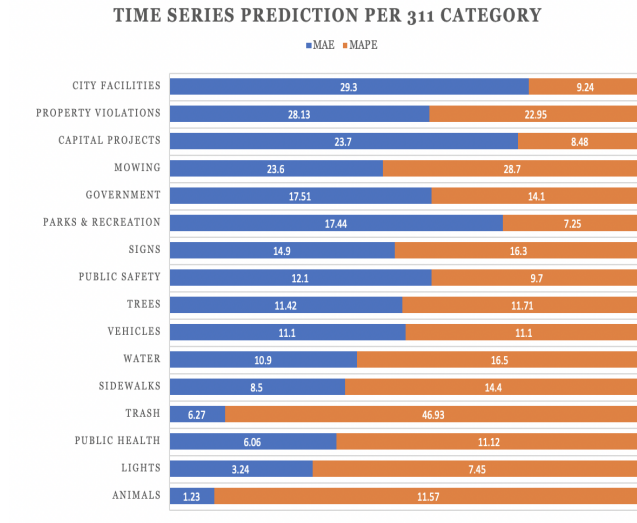


Figure 41: 311 Call Time Series Validation

requests in Kansas City.

**Evaluation:** Figure 41 shows the evaluation of time series results on a 70-30 split of data. The time series model for the 311 service data, which is a nonlinear regression based model for 311 call response times, was evaluated: (1) MAE (Mean Average Error) (defined in Eq. 4.11) is the measure of the difference of predicted versus observed, (2) MAPE (Mean Average Percentage Error) (defined in Eq. 4.12) is a measure of prediction accuracy of the forecasting (loss function for regression in machine learning).

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - x_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n} \quad (4.11)$$

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (4.12)$$

**Results of the Time-series Model** Figure 41 shows the MAE and MAPE scores as the

Table 9: 311 Time Series Predictive Model Accuracy

Model Name	MAE	MAPE
LSTM	9.71	105.63
LSTM + Window	7.81	79.33
LSTM + Time Steps	5.70	65.87
LSTM + Memory	7.44	84.50
Stacked LSTM + Memory	7.84	93.62
DCNN (SeriesNet)	14.20	210.32
Prophet (Proposed)	<b>3.45</b>	<b>14.38</b>

performance of the 311-time series model. The best classes of the response time prediction are *Animal* and *Tracy* with 1.23% and 6.27% of its MAE. The worst class accuracy is *City Facility* with 29.3% of MAE. In terms of MAPE, the best class is *Lights* with 7.45% of MAPE, and the worst type is *Tracy*, with 46.93% of MAPE. Thus, the overall accuracy for the response time estimation is about 70%. The Prophet forecasting model showed superior performance when compared to the deep learning models (Table 9).

**Discussion of the Time-series Mode** The time-series model shows a steadily decreasing trend overtimes, which suggests an improvement in the performance of the city service. The cases that were used to take around 80 days to resolve in 2015 were solved in under 20 hours in 2020. The most significant improvement was “Animals”, “Government”, “Mowing”, “Vehicles”, and “Water”. “Capital Projects,” “Public Safety,” “Sidewalks,” “Signs,” and “Trees” have a slower and flatter downward trend. “City Facilities,” “Dangerous Building,” and “Lights” exhibit strong seasonality, without a significant downward or upward trend. “Property Violations,” “Public Health,” “Street,” and “Trash” show highly idiosyncratic patterns; the outbreaks of the problems rather than seasonality influences

service time. These are the most problematic areas of 311 service delivery.

#### 4.1.11 “What” Model: custom topic model Clustering Modeling

**Data:** All 311 text records were used. The goal of “What” Model is inductively to summarize the main themes of residents’ complaints and then analyze the relationship between the main themes/topics of complaints with the internal-facing 311 service categories. We achieved this goal by (1) summarizing the main themes/topics of the complaints by using LDA topic modeling; (2) clustering documents into topic categories by using the Balanced LDA+BERT Clustering (custom topic model) model; and (3) visualizing the relationship between themes/topics and internal-facing 311 service categories.

Using that information, we made document clusters belonging to different themes. This process can be particularly useful when people attempt to simplify existing categorization systems. As the process is unsupervised and fully automatic, new categorizations (often simplified) can be easily generated without laborious human tagging.

**Training the LDA Topic Model:** The preprocessing includes text cleaning, tokenization, stop-word removal, and lemmatization. To ensure optimal results, coherence scores were calculated. Figure 42 shows the coherence values across a range of 8 to 25 topics, and 12 topics had the highest topic coherence (0.5512). 17 topics (the existing 311 service categories have 17 categories) had a coherence score of 0.5350. In the experiments, we demonstrated that the quality of the LDA topic modeling had a significant impact on the custom topic model clustering models.

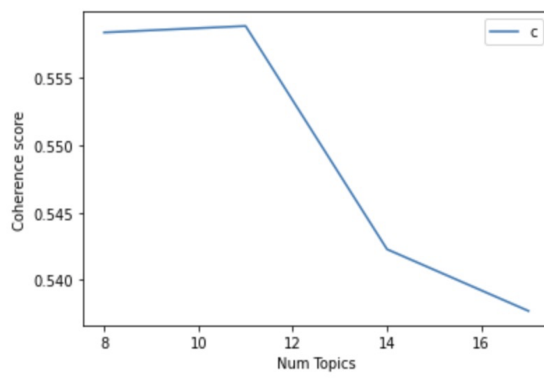


Figure 42: Coherence for Optimal Topic Modeling

Table 10: Top 12 Topics and Topic Terms

ID	Topic	Topic Terms (10)
Topic 1	Traffic	traffic, corner, hydrant, lane, hole, fire, street, north, road, south
Topic 2	Animals	dog, black, taker, white, brown, pit, large, small, loose, flat
Topic 3	Limbs and lights	limb, dead, street, light, note, cartograph, replace, pole, possible, soon
Topic 4	Case referred	refer, please, case, see, note, work, still, service, closed, missing
Topic 5	Trucks	truck, get, city, last, week, time, disabled, put, state, know
Topic 6	Water leak	water, leak, meter, home, pressure, low, coming, pipeline, street, state
Topic 7	Bulky items	property, item, bulky, recycle, leaking, neighborhood, dumping, sewer, maintenance, appointment
Topic 8	Vehicles	vehicle, car, parked, street, sign, plate, abandoned, parking, front, month
Topic 9	Trees and pot-holes	tree, street, city, need, side, pothole, sidewalk, large, right, removed
Topic 10	Trash	trash, missed, bag, violation, sticker, picked, recycling, collected, block, time
Topic 11	Houses	house, front, tire, open, door, building, damaged, side, home, vacant
Topic 12	Property maintenance	yard, property, grass, weed, lot, back, need, cut, tree, debris

**Results of the LDA Topic Model:** Table 10 showed the top 10 words of the 12 topics from the LDA model. We conducted a topic-by-topic qualitative comparison of the 12-topic and the 17-topic LDA models since the internal-facing 311 service categories have 17 categories. Both the 12-topic and the 17-topic LDA models identified “traffic,” “animals,” “limbs and lights,” “water leak,” “bulky items,” “vehicles,” “trees and potholes,” “houses,” and “property maintenance.” Topic 10 “Trash” of the 12-topic model was split into three topics in the 17-topic model. The 17-topic model had two topic categories that were hard to interpret. The top 10 words of these unclear themes were: 1) get, people, need, time, cover, American, ice, concern, state, area; 2) meter, refer, see, please, note, traveler, turned, construction, flat, driveway. Within the 12 topics, Topic 5 Trucks appeared to be a bit challenging to interpret. This comparison showed that the 12-topic model summarized the themes better than the 17-topic for a human to interpret.

**“What” Model - Topic Modelling:** Our Custom Topic Modelling model is different from the existing LDA+BERT model since it balances the LDA topic and BERT document vectors. We demonstrated the importance of balancing the LDA topic vector and the BERT document embedding by experimenting with the length of  $W'$ , and  $H'$ . A balanced length of the vectors performed the best and produced the highest Silhouette score (see Table 11). We demonstrated the impact of the LDA topic model (coherence) on the clustering model by using the 17 topic vector in the same autoencoder. 12 topics were of higher quality than 17 topics and thus produced better results of clustering.

**Discussion of the custom topic model Model:** Tables 14 - 16 show this relationship. The cross-tabulation tables include the 17 service categories (including the “other” and

Table 11: Silhouette Scores (SS) of Topic Models

Models	Dimension of W	#Clusters	W'/H' Ratio	SS Score
custom topic model (Ours)	12	12	<b>32:32</b>	<b>0.3623</b>
LDA	12	12	12:52	0.2985
BERT+LDA	17	12	17:47	0.2405
TF-IDF	17	17	17:47	0.2321
Baseline [95]	12	12	N/A	0.2745

the “data not available” categories). The counts of the documents per topic/theme were based upon the membership from the custom topic model clustering model. The 12 main topics/themes of complaints were concentrated on the top categories of the internal-facing 311 service categories. “Trash,” “Water Leak,” and “Animals” were three major problems that both categorization systems agreed upon. The “Traffic” problem involved both the lights/signals/signs service and the street/sidewalks service. The “Limbs and Lights” problem was largely a public safety concern. The problems of “Houses” and “Property Maintenance” involved services of property violations and street/sidewalks. Table 14(a) showed that a majority of the complaints in the “other” category belonged to the themes/topics of “Trees and Potholes,” “Property Maintenance,” and “Bulky Items.” The “other” category is often created to group items that are hard for human operators to put them into a category.

The themes identified by inductive categorization overlaps with the existing internal-facing 311 categorization. Both categorizations have “Property violations,” “Street,” “Animals,” “Trash,” “Trees,” and “vehicles.” The inductive categorization revealed complaint



categories that were not part of the existing internal-facing 311 categorization: “Neighborhoods,” “Houses and yards,” “Common areas,” “Potholes and roads,” and “Trash in the neighborhood.” These categories represent major types of concerns that may be sub-categories of existing 311 categories.

The inductive topic model highlighted these main concerns from the resident’s perspective. For example, “Potholes and roads” highlights the concerns over potholes. There are also many neighborhood issues, including complaints about the “Common areas” and “Houses and yards” of the neighborhood, “Trash in neighborhoods,” and other types of complaints in the “neighborhoods.” Figure 14 (in Chapter 3), shows the identification of the ten dominant topics with the corresponding keywords for each topic. *Animals* and *Mowing* are the top two dominant topics with the topic contributions of 0.1271 and 0.1274, respectively.

The inductive categorization ( $N = 12$ ) had fewer categories than the existing 311’s internal facing categorization ( $N = 17$ ). Figure 34 shows the mapping process from a call description to the internal and external categorization. For example, the first description (“There is a Caucasian man...”) was first processed by Model 1’s department prediction, and mapped to the Health Department. Then, Model 1’s 311 categorization put it into the category of Public Health (the Artificial Process Intelligence will recommend the category “Public Health to the Operator). Meanwhile, Model 3 predicts that it will take about 21.55 days to solve the problem. Model 2, using the probability of each topic, determines that this description belongs to the complaint category of “Neighborhood” from the residents’ perspective. Mapping residents complaints onto the existing 311 categorization (We can

see an even distribution of topics - signifying the use of 311 for a variety of complaints.

The map shows that the complaints were distributed evenly across all 12 internal-facing categories. Four overlapped categories are among the top 12, suggesting “Property violations,” “Animals,” “Trash,” and “Trees” are shared concerns of both the residents and the city management. The “Tell More” LDA+BERT model created connections between the internal- and the external-facing 311 categorizations. The external-facing inductive categorization facilitates the communication between the human operator and the residents, and identification of the internal-facing categories and departments. The “Tell More” LDA+BERT model can help operator to quickly classify a call into one of the dominant topic categories.

Table 10 showed the top 10 words of the 12 topics from the LDA model. We conducted a topic-by-topic comparison of the 12-topic and the 17-topic LDA models since the internal-facing 311 service categories have 17 categories. Both the 12-topic and the 17-topic LDA models identified “traffic,” “animals,” “limbs and lights,” “water leak,” “bulky items,” “vehicles,” “trees and potholes,” “houses,” and “property maintenance.” Topic 10 “Trash” of the 12-topic model was split into three topics in the 17-topic model. The 17-topic model had two topic categories that were hard to interpret.

The top 10 words of these unclear themes were: 1) get, people, need, time, cover, american, ice, concern, state, area; 2) meter, refer, see, please, note, traveller, turned, construction, flat, driveway. Within the 12 topics, Topic 5 Trucks appeared to be a bit challenging to interpret. This comparison showed that the 12-topic model summarized the themes better than the 17-topic for human to interpret.



**Training of custom topic model:** Text cleaning was conducted, and sentence BERT embeddings were generated using pre-trained uncased BERT. The custom topic model was trained for 5 epochs. The encoder was saved and used to generate hidden layer vector representations. We conducted four experiments to combine the topic per document probabilities and the BERT document embedding.

The baseline model replicated the existing LDA+BERT model: The LDA topic vector  $W$  (from the 12-topic LDA model) was concatenated with the BERT document vector  $H$ . This joint vector  $T$  was entered into a shallow autoencoder with one dense layer that reduced the dimension of  $T$  to 32. The hidden layer vector  $t$  ( $D = 32$ ) was entered into a Kmean cluster model, to cluster the documents into 12 categories.

The problem with the existing LDA+BERT clustering model is that the results rely on the BERT document embedding more so than the LDA topic vector due to the large difference of the vectors' length. As a result, it fails to combine the strength of LDA topic modeling and the pre-trained BERT. Thus, we modified the autoencoder to allow LDA topic vector and BERT document embedding to learn separately (see Figure 12 in Chapter 3). The custom topic model model was trained for 5 epochs.

We demonstrated the importance of balancing the LDA topic vector and the BERT document embedding by experimenting with  $W'$ , the dense layer that encodes the LDA topic vector, and  $H'$ , the dense layer that encodes the BERT document embedding in the autoencoder (see the Imbalanced LDA+BERT Clustering models (ILBC) in Figure 42). We demonstrated the impact of the LDA topic model (coherence) on the clustering model by using the 17 topic vector in the same autoencoder (see 17-Topics in Figure 42). By

comparing the Silhouette scores of the clustering models, we concluded that the Balanced LDA+BERT Clustering model (custom topic model) performed better than the existing LDA+BERT and ILBC clustering models. Each model was trained for 5 epochs.

**Evaluation of custom topic model:** The objective function of custom topic model was log-cosh. The log-cosh loss function is a regression loss function that behaves similarly to the mean squared loss but is robust to outliers. It is the logarithm of the hyperbolic cosine of the prediction error. Formally:

$$L(y, y^p) = \sum_{i=1}^n \log(\cosh(y_i^p - y_i)) \quad (4.13)$$

We conducted four experiments to compare custom topic model with other models. First, we compared custom topic model with the baseline model. First, the baseline model replicated the existing LDA+BERT model: The LDA topic vector  $W$  (from the 12-topic LDA model) was concatenated with the BERT document vector  $H$ . Then, we conducted four experiments to combine the topic per document probabilities and the BERT document embedding. The baseline model replicated the existing LDA+BERT model: The LDA topic vector  $W$  (from the 12-topic LDA model) was concatenated with the BERT document vector  $H$ . This joint vector  $T$  was entered into a shallow autoencoder with one dense layer that reduced the dimension of  $T$  to 32. The hidden layer vector  $t$  ( $d = 32$ ) was entered into a cluster model, to cluster the documents into 12 categories.

The second comparison was to experiment with different dimensions of  $W'$  and  $H'$ : 1)  $d'_W = 12$ ,  $d'_H = 52$ , and the number of clusters  $N = 12$ ; 2)  $d'_W = 17$ ,  $d'_H = 47$ ,  $N = 12$ . The third comparison was to compare different LDA topic vectors: 17-topic

vector vs. 12-topic vector:  $d'_W = 17$ ,  $d'_H = 47$ ,  $N = 17$ .

All models were trained for 5 epochs. The latent representations of the proposed model custom topic model, the original LDA+BERT model (baseline), were entered into the same K-means clustering algorithm.

We evaluated the clustering models using using the log-cosh loss function (80-20 train-validation ratio), the Silhouette score and the elbow method. The Silhouette score measures how coherence a document is to its own cluster compared to the other clusters. It has a range of  $[-1, 1]$ , with 1 indicating high cohesion within the cluster and high separation from the other clusters. Thus, it is a better measure than visualization when dealing with higher dimension clustering. The elbow method is a visual method to identify the optimal number of clusters by plotting WCSS (Within-Cluster Sum of Square). Putting it altogether, the results from the above models are presented in Figure 44.

#### 4.1.12 “Why” & “How” Model: Question Answering

To prove our hypothesis that our model, which is based on the models built with 311 call descriptions that produce predictive results. Answering questions based on the predicted outcomes could not be expected from the ALBERT-based question answering model. The answer to the question of “What Happened?” is informed by predictive results from the predictive model shown in Figure 38 (a). The answer to the question of “Which department handles this issue?” is informed by predictive results from the predictive model shown in Figure 37 (b). The answer to the question of “How long will it






Complaints	Department	311 Internal	When (days)	Tell more?	311 External
There is a caucasian man building a lean-to behind the dumpster adjacent to my property. The dumpster is in the parking lot of what was 'Bonds Chicken and Blues', which is now vacant. I am concerned about safety and public health.	Health	Public Health	21.55	['property', 32.43], ['attach', 13.51], ['issu', 10.81], ['grass', 10.81], ['junk', 10.81], ['truck', 8.11], ['yesterday', 5.41], ['abandon', 2.7], ['mailbu', 2.7], ['trailer', 2.7]]	
Citizen is reporting several wrecked vehicles parked on the street, where school buses cannot safely navigate down the roadway due to all the wrecked vehicles in front of the business parked on the street.	Public Works	Street	23.75	['street', 38.24], ['time', 17.65], ['bin', 11.76], ['peopl', 5.88], ['bridg', 5.88], ['supervisor', 5.88], ['note', 5.88], ['train', 2.94], ['gregori', 2.94], ['driveway', 2.94]]	
Citizen is reporting multiple vehicles that are unlicensed and are disabled and have not moved in weeks, that are parked on the street and preventing vehicular traffic from safely navigating thru the street.	KCPD	Traffic	24.75	['traffic', 25.93], ['recycl', 14.81], ['car', 14.81], ['sign', 11.11], ['locat', 11.11], ['bhd', 7.41], ['cater', 3.7], ['hair', 3.7], ['build', 3.7], ['west', 3.7]]	
Citizen is a mail carrier and he reporting a large white pitbull loose in the area. Also a small sheppard mixed. These dogs are keeping him from delivering the mail. Pitbull aggressive, sheppard mix non-aggressive. Pitbull has chain. Sheppard mix has no collar. 67	NHS	Animals	16.91	['week', 19.23], ['dog', 15.38], ['window', 11.54], ['hazard', 11.54], ['busi', 11.54], ['glass', 7.69], ['jackson', 7.69], ['anim', 7.69], ['meyer', 3.85], ['root', 3.85]]	
Citizen states that it smells like sewer outside the home. Specifically there is a small hole near her drive way but it is right before the sidewalk and the odor seems to be coming from there.	Parks and Rec	Sidewalk	13.69	['lane', 29.41], ['side', 14.71], ['home', 14.71], ['barri', 8.82], ['sidewalk', 8.82], ['charlott', 5.88], ['barrier', 5.88], ['week', 5.88], ['trail', 2.94], ['forward', 2.94]]	

Figure 44: 311 - Topic Modelling

take?” (estimating response time) is informed by predictive results the time series forecasting model shown in Figure 18. The answer to the question of “Why & How?” is informed by the predictive results from the topic identification model shown in Figure 15.



Table 12: AI-Community-311-Question-Answers

Description	Who?	Where?	Topics?	When?
<i>Citizen is calling to report a sewer back up in home. Water is murky and stinks and the pipe needs to be flushed out because it is getting in the house.</i>	Water Services	Water	Topic 4: '0.167*'water'' + 0.125*'pot-hole'' + 0.083*'yard'' + 0.062*'to-day'' + '0.062*'state'' + 0.042*'tenant'' + 0.042*'plate'' + 0.042*'hill'' + '0.021*'wagon'' + 0.021*'overgrowth''	10 days
<i>Citizen is calling to report that there is a leak in the street and it is flooding out water. right at this intersection.Needs to be fixed.</i>	Parks and Rec	Trees	Topic 3: 0.115*'tree'' + 0.077*'street'' + 0.077*'call'' + 0.058*'intersect'' + '0.058*'house'' + 0.038*'ter'' + 0.038*'taker'' + 0.038*'debris'' + 0.038*'trash'' + 0.019*'bull''	11 days

#### 4.1.12.1 Validation

The KCMO 311 service data [43] is split into 80-20 train-validation ratio to in order to evaluate model's performance. The current review of the data belongs to the Kansas City region, where we can evaluate the role being played by multiple departments in the region to make the neighborhood a better place. As seen in Figure 37, the total numbers of 311 service categories and departments are 17 and 15, respectively.

Below are the departments involved in 311 calls. NHS, City Managers Office, Public Works, Water Services, Parks and Rec, Northland, City Planning and Development, Health, KCPD, South, Finance, Aviation, Convention and Entertainment Center, Fire, General Service, Northeast, Parks and Rec, NCS, Information Technology, Parks and Recreation, Housing Community Dev, IT and Municipal Court. Validating the 311 results followed corresponding scores like MCC, Accuracy, Precision and an ensemble of the GLUE benchmark scores. The MAE and MAPE scores as used the performance metrics of the 311 time series model. The best classes of the response time prediction are *Animal* and *Tracy* with 1.23% and 6.27% of its MAE.

The MAPE metric is also sometimes reported as a percentage, which is the above equation multiplied by 100. The difference between  $A_t$  and  $F_t$  is divided by the Actual value  $A_t$  again. This calculation's absolute value is summed for every forecast point in time and divided by the number of fitted points  $n$ . Multiplying by 100% makes it a percentage error. The worst class accuracy is *City Facility* with 29.3% of MAE. In terms of MAPE, the best class is *Lights* with 7.45% of MAPE, and the worst type is *Tracy*, with 46.93% of MAPE. The overall accuracy for the response time estimation is about 70%.

Table 13: Technologies used in StoryNet modeling

Keras	A data flow and differential programming library for machine learning
TensorFlow	A data flow and differential programming library for machine learning
DNN	Deep Neural Networks - A generalized mathematical model for data classification
SciKit Learn	A library to work with classification, regression and clustering algorithms
Gensim	A modern statistical machine learning python library for topic modeling and natural language processing (NLP).
NLTK	The Natural Language Toolkit, to process the text data.
Spacy	Python library for advanced NLP tasks.
Pandas	A open source data analysis and manipulation tool

The Prophet forecasting model showed superior performance when compared to the deep learning models. Putting them all together shows the advantage of StoryNet with a small example in Figure 45.

Table 14: Topics and Dominant 311 Service Categories

	Animals	Lights, Signals, Signs	Property Violations	Public Health	Public Safety	Street, Side-walks	Trash	Vehicles	Water	Other
Traffic	1049	5602	319	46	984	7501	1334	79	2460	563
Animals	20748	38	73	177	30	45	139	41	27	10
Limbs & Lights	3581	4545	466	3	9055	1649	2120	10	116	306
Case Referred	735	1070	2384	674	871	1919	2821	156	4357	1043
Trucks	1025	1274	2560	3455	1498	3688	8692	972	2442	1242
Water Leak	3337	36	1173	715	84	903	492	130	22682	357
Bulky Items	59	72	5887	79	352	430	8603	204	252	3475
Vehicles	218	3495	2059	31	155	1235	748	9864	1161	306
Trees, Potholes, Limbs, Lights	650	767	3090	53	3216	13120	2349	79	1289	5212
Trash	428	18	1124	15	38	120	46120	6	44	269
Houses	1462	138	6503	558	433	861	2520	149	562	547
Property Maintenance	765	423	7616	346	2602	2154	5942	538	2378	4977

Table 15: 311 Service Categories Examples

Traffic	Citizen calling to report the traffic lights going north and south are stuck on red and going east and west the lights are stuck on green. Red Bridge and Hickman Mills Dr.
Animals	The citizen is reporting a dog bite that happened at this address. The incident happened on 06/15/16 between 6:30p and 7:30p The bit a human victim and another dog. The attacking dog is white with black.
Limbs & Lights	Citizen is reporting two street lights out. The two pole numbers are SDM1010 and SDM1011.
Case Referred	The citizen is requesting a callback in regards to case number 2015072174. The note from 7/9/2015 states that a letter was mailed to the citizen on 6/5/2015 and the citizen has not received the letter.
Trucks	No snow plow or salt truck has been down the street. I know deadends are last but it said to wait 36 hours and it has been longer than that. Thank You.
Water Leak	Citizen called to report water leak. Water leak is located in the street around this address. Water is clear and odorless. Water is trickling from both sides of the pavement.
Bulky Items	Citizen s requesting bulky appointment but address is not in the scheduler.
Veehicles	Citizen is calling to report an abandoned white Chevy van sitting in front of this location. The van has been sitting on the street for months.
(1) Trees, Potholes, (2) Limbs, Lights	(1) The caller is reporting two ROW trees located at the curb be removed due to the roots of the tree are buckling his sidewalks. (2) Citizen reports city oak trees located on the right of way along W 69th Ter and Ward parkway need to be trimmed.
Trash	Citizen called to report had 2 bags of trash out said that the trash truck collected 1 bag and left the other, also did not collect neighbors trash.
Houses	The citizen is calling to report that the house is open to entry. The doors and the windows are all off.
Property Maintenance	The citizen reports the grass and weeds are overgrown in the front and back yard. There is brush in the yard. The yard has not been mowed in over 2 months now.

Table 16: Topics and Minor 311 Service Categories

	Parking	Parks & Recreation	City Facilities	Legal	NA	Maintenance	Neighborhood	Noise	Traffic
Traffic	100	381	14	335	3	4	0	0	124
Animals	1	28	2	5	2	0	0	0	0
Limbs & Lights	10	58	3	33	3	1	0	0	1
Case Referred	43	232	41	207	10	10	21	3	18
Trucks	42	407	51	300	13	2	25	59	36
Water Leak	28	62	14	17	5	2	1	1	1
Bulky Items	12	56	8	40	19	9	1	0	0
Veehicles	34	70	6	94	9	0	4	0	151
(1) Trees/Potholes	102	308	15	275	20	12	0	0	16
(2) Limbs/Lights									
Trash	10	42	0	18	6	0	0	0	0
Houses	23	132	12	42	6	1	4	0	2
Property Maintenance	88	878	23	140	35	16	7	1	7

We can see that there are stories (or 311 reports) that are talking about missing recyclables at different points in time.

## 4.2 Case Study 2: OCEL.AI

OCEL.AI demonstrates the process of human computer collaboration in the process of AI and machine learning starting from real world story telling to story about findings and observations from the collaboration process.

Topic models may provide additional signals for semantic similarity, as earlier feature-engineered models for semantic similarity detection successfully incorporated topics ([77], [99], [59], [111]). They could be especially useful for dealing with domain-specific language since topic models have been exploited for domain adaptation ([40], [35]). Moreover, recent work on neural architectures has shown that the integration of topics can yield improvements in other tasks such as language modeling [32], machine translation [17], and summarisation ([64], [106]).

The conceptualization of this project is based upon OCEL.AI. OCEL.AI is a data science research approach to and design thinking of complex problem-solving, modeling, and experimentation that often involves multiple entities and factors. This approach utilizes domain knowledge of human communication (including interpersonal, organizational, and mass media communication) to create story schema for the design of AI processes and interfaces. The AI process and interface for community and neighborhood issues feature “what”-initiated and solution-oriented problem-solving. By using the “5W + H” journalistic storyline, the system analyzes 311 calls, emails, and tweets from Kansas

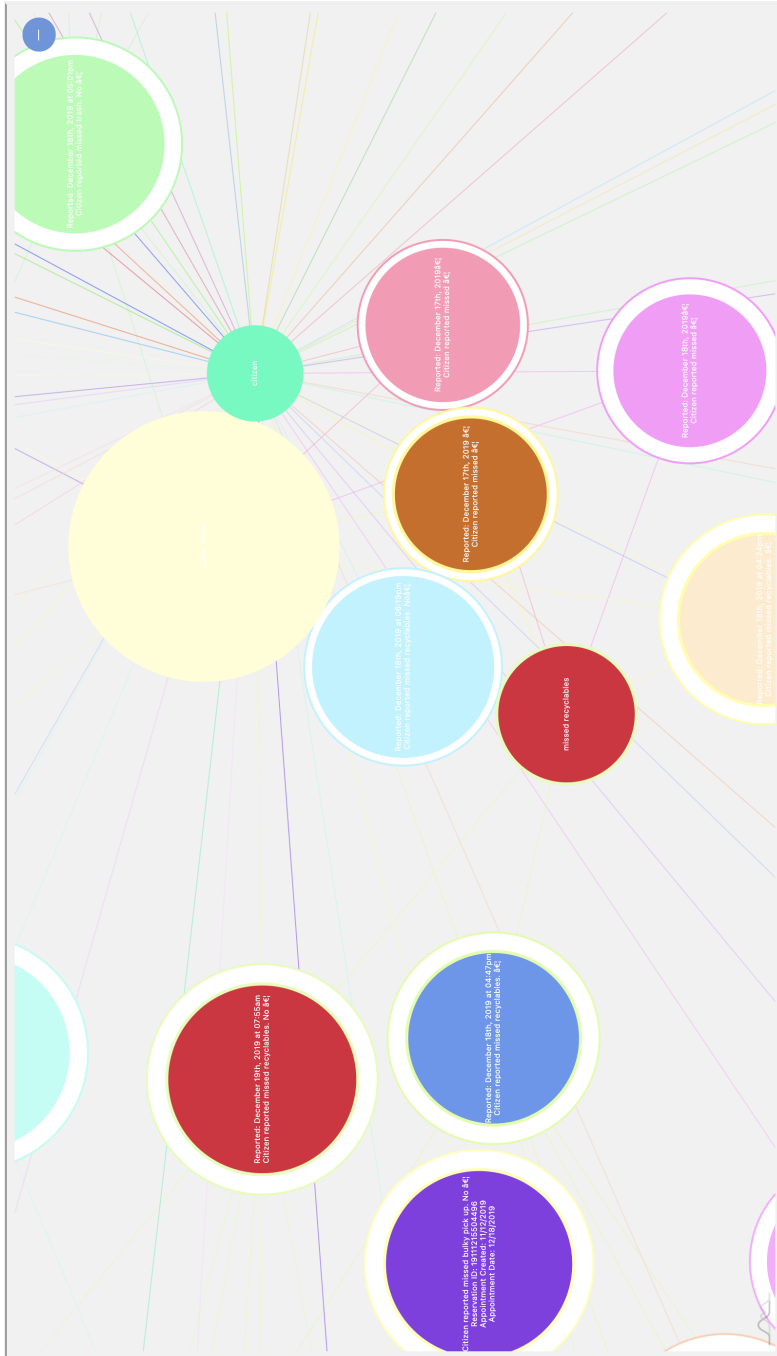


Figure 45: StoryNet - Advantage



City and other relevant data sets, generates process intelligence, and facilitates solutions by following these steps of analysis in order: 1) what happened (what the community issue is); 2) where it happened (the location, environment, and situation where city service is needed); 3) when it happened; 4) who the community is; 5) why and how it happened (various causes and consequences).

This process is initiated by the what question, analyze possible causes and consequences, and makes evaluative predictions of potential solutions. The case study demonstrates short-term AI solutions. And, a conceptual framework of predicting long-term solutions is presented, which involves policy changes, science studies, and community actions.

Using deep learning models and the OCEL.AI concept, we proposed a new question answering framework, called *AI-Community-311*, aiming to dynamically answer questions by connecting underlying deep learning models using real-world data. The contributions of this thesis are summarized as follows: 1) The community-in-the-loop approach was realized via deep learning models that can understand what the residences say about the community. Using the OCEL.AI's design thinking, we trained deep learning models on the 311 calls from the communities. We mainly focus on presenting problems in terms of (i) what happened (the kind of 311 problems), (ii) who can handle it (311 call department), (iii) when it happened, (iv) where it happened. 2) We have designed an AI-based question answering framework, 311-Chatbot Q&A, by building a sequence of predictive models for the 311 service domain: (i) a predictive model for determining the category for the description of a 311 service, (ii) a predictive model for determining the

working department for described a 311 service, (iii) a predictive model for forecasting time for solving the 311 service request, and (iv) a predictive model for identifying the topics of the 311 request. 3) The comprehensive evaluation has been conducted for the 311-Chatbot Q&A by leveraging the state-of-the-art deep learning question answering, ALBERT. Also, individual components of the 311-Chatbot Q&A have been evaluated.

OCEL.AI acknowledges the non-linear progression of complex problem-solving, modeling, and experimentation that often involve multiple entities and factors. Therefore, a significant contribution of OCEL.AI is creating a linear structure of narratives that human storytellers are familiar with to facilitate non-linear human-machine problem-solving. Accordingly, OCEL.AI integrates five components of data science into a linear structure of storytelling.

The linear structure of the story-driven experiential learning has five chapters: 1) life, 2) data, 3) the scientist and AI, 4) users, and 5) the society. Borrowing from data journalism, our storyline uses “5W + H” to guide learning: Who is it about? What happened? When did it take place? Where did it take place? Why did it happen? How did it happen? For each chapter, we designed a story framework to guide learners to create their own stories (assignments). All five chapters form a feedback loop, which allows adjustment and optimization. Two calibrations are also added to highlight the importance of experimentation and alignment.

Ammanabrolu et al. [2] use knowledge graph to guide users to fill in important information using thematic knowledge learned from similar stories. In this study, interesting, they demonstrated the human-machine collaboration using deep neural model

and a rules-based baseline for knowledge graph construction, story generation and game formulation.

The first wave of science studies in 1950s and 1960s has proven that science is not able to solve social problems (community problems are essentially social) without two-way communication and a collaborative process of co-creation with the communities [22]. Communities have long-time first-hand experience with community issues, which should receive equal or comparable weight via delegation (like elected public servants) or representation (like opinion polls and surveys) [22].

Thus, community engagement has become an important component of the second and third wave of science studies; this trend can be reflected in the growing emphasis on community engagement in many NSF grants. A distinctive feature of the second and third wave of science studies is to admit the uncertainty of scientific knowledge and the lack of scientific consensus. Then, given these assumptions, How can scientists and communities work together to solve community problems?

Thus, today's challenge is not to argue for the importance of community-in-the-loop but to create a mechanism to address the methodological question of "how" to bring communities in the loop. A recent study funded by the UK's Economic and Social Research Council spent three years examining how autism research could become more participatory, i.e., relevant and transformative to the autism community [31]. The findings revealed that the major obstacles were lack of supportive components [8] for communities to get heard and involved, and a mechanism that allows multi-directional communication among all stakeholders for constructive dialogues.

To address urban issues, scholars argued that engineering models and urban design should make an effort to understand community concerns, aspirations, and adaptation ideas, and explore and evaluate solutions in terms of its responsiveness to the local context and economic viability [83]. The case study on urban flood resilience in Australia documented the use of visualization techniques to facilitate the process of community visioning participated by community stakeholders [83]. Community visioning provided insight on community needs and wants, and then guided engineering modeling and analysis and urban design [83]. Thus, the first and foremost step of a community-in-the-loop approach is to teach machines how to listen to community inputs and understand human insight. Realized mainly via workshops, the case study demonstrated community-driven problem-solving in flood risk governance [83].

Similar trends of empowering human are witnessed in computer science, engineering, and AI design. Schneider et al. [87] pointed out the key to effective design that differentiates effective from ineffective approaches to empowering humans is to form a clear understanding of Schneider et al. [87] presented the empowerment in the Human-Computer Interface (HCI) by the research community.

They pointed out the need for more effective design guidelines and best practices, including "needed as a foundation for design guides and best practices and to differentiate between effective and ineffective approaches of empowerment. In order to develop such metrics, a clear understanding of empowerment is needed: "Who is the target group, which are the targeted psychological components?" In particular, they identified the increasingly unclear use of "user experience" in a variety of usability-related research (e.g.,

enabling technologies or on choice architecture [41]).

Additionally, the complexity of community problem-solving requires not only one-time input from the community at the onset but also iterative and multi-directional communication, feedback, and optimization to sustain long-term problem-solving.

To do so, this study presents story-driven design thinking called OCEL.AI (Open Collaborative Experiential Learning.AI). OCEL.AI uses “5W + H” story elements as a two-way communication and sense-making tool to engage the community in defining and solving the problem. Human beings are storytelling animals, and narratives (stories) are the means by which we make sense of, organize, and understand the world [60]. Thus, stories/narratives are a genre of discourse that is mutually understood by communities with experiential expertise, researchers with domain expertise, and AI designed using a story-driven approach.

The proposed model on the 311 call domain in the OCEL.AI story framework will be generated using the latest NLP and deep learning research, including BERT [24], ALBERT [47] that show promising results for various NLP applications. Figure 46 shows the Neighborhood Story Framework.

- **Who:** (1) A Computer Science Master’s student (“I”); (2) AI: The machine learning models built using the data relevant to the life story; (3) Residents in KC who are looking for a solution for their problem; (4) City department who can provide solutions to Residents; (5) Neighborhoods who are living near to the residents.
- **What happened:** predictive models to identify the city department who can handle their problems. (1) Classification Models to show the topics from the 311 calls in

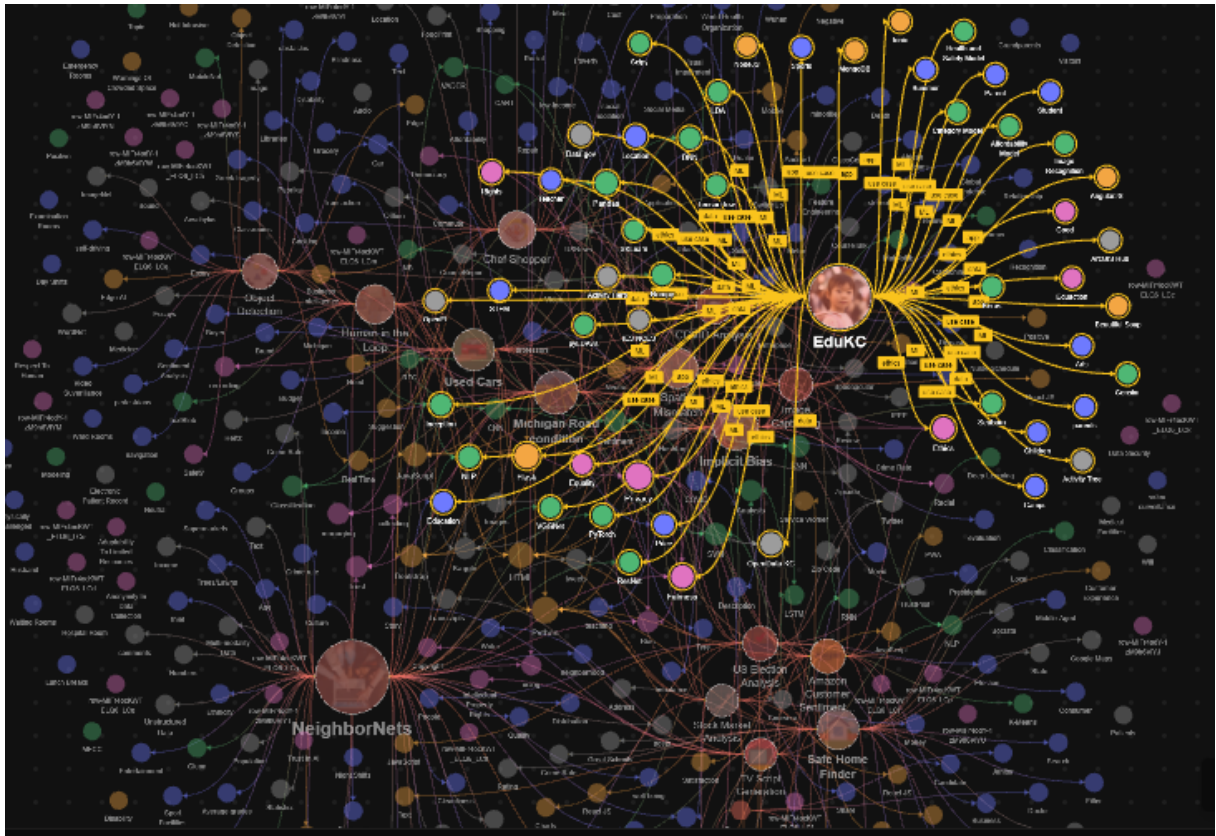


Figure 46: OCEL.AI StoryNet Design

Kansas City. (2) Time Predictive Models to predict how long does it take to handle the complaint. (3) Work Predictive Models to predict the department which can handle the complaint. (4) Causal Models to determine the cause of the problem. (5) Consequence Models to determine the consequence of the problem. (6) Prioritization Model to compute the cost and effort of the problems and determine the priority of tasks.

- **When & where:** predictive models to estimate the time to respond to it. The models will be built using the data that are used considering the time.
- **Where:** The ML models will be built by select data considering the locations of all available
- **Who:** the skills and expertise of community members; streets, wifi, pipes, etc.; churches, non-profits, HOAs, etc.; businesses, etc. *(1) A Computer Science Master's student ("I"); (2) AI: The machine learning models built using the data relevant to the life story; (3) Residents in KC who are looking for a solution for their problem; (4) City department who can provide solutions to Residents; (5) Neighborhoods who are living near to the residents.*
- **Why & How:** predictive models for possible causes and consequences.

#### 4.2.1 What, Where and Why from QA

To generate context-aware conversation from the description provided by 311, we used the ALBERT QA model. First, recorded 311 calls are processed ("comprehended")

via deep learning models, to generate knowledge for what the problem is, when and where it happened, and why the problem arises. Second, the knowledge from the predictive models are added as responses to the knowledge graph of the Q&A model.

To further aid the user, the conversation is designed to focus on the problem, giving a quick and accurate view of the case. Table 17 shows some examples of the 311 call descriptions and questions. In the following section, we present predictive models

Table 17: ALBERT’s Question Answering in 311 Services

<b>Description</b>			
<i>Case 1: Citizen is reporting dumping of trash and large items at the curb is a nuisance and is spilling out on to the sidewalk. Citizen did not specify exactly the items that are being dumped.</i>			
<b>Who</b>	<b>What</b>	<b>Where</b>	<b>Why</b>
Citizen	nuisance	on the sidewalk	dumping of trash and large items at the curb
<b>Description</b>			
<i>Case 2: Where the alley (between Belleview and Jarboe) meets W 45th street, the asphalt is crumbling. Cars can not go up or down the hill due to the unevenness. Cars are being scraped on the bottom. Repair other than asphalt is needed.</i>			
<b>Who</b>	<b>What</b>	<b>Where</b>	<b>Why</b>
Citizen	Cars can not go up or down the hill	W 45th street	asphalt is crumbling
<b>Description</b>			
<i>Case 3: Citizen is reporting a small water tower at this location that is leaking . Citizen reports grass is wet all around it and excess water is flowing into the storm drain.</i>			
<b>Who</b>	<b>What</b>	<b>Where</b>	<b>Why</b>
Citizen	grass is wet and excess water is flowing into the drain	a small water tower	a small water tower is leaking



that address what, when, and where. Future studies are needed to further address who, why, and how.

Stories are also the main subject of investigation in deep learning and knowledge graph. Ammanabrolu et al. [2] use a knowledge graph to guide users to fill in necessary information using thematic knowledge learned from similar stories. They demonstrated the human-machine collaboration using a deep neural model and a rules-based baseline for knowledge graph construction, story generation, and game formulation.

Along those lines, OCEL.AI storytelling not only serves the goals of data visualization but also theorizes and problem-solving. Data visualization can be described as a process of using information visualization techniques and narrative storytelling to achieve communication goals [27]. The purposes of data visualization are using data to inform, educate, advocate, influence, and persuade audiences. In contrast, the purpose of OCEL.AI storytelling is to serve the goals of data visualization and theorizing, and problem-solving. Ochs et al. pointed out that storytelling is a powerful tool for complex theory-building [69].

Axelrod and Kahn argued that storytelling goes hand-in-hand with modeling and can establish links between personal and social worlds via meaningful interpretation of large-scale datasets [4]. Following this school of pedagogical thinking, our team designed five stages of end-to-end story-driven data science/AI experiential learning, which entails data visualization. Our work is also influenced by science stories [33] and visualization ethics [23]. Science stories emphasize the importance of using scientists' discovery,

rescue, and mystery stories to engage more people into problem-solving [33]. And, visualization ethics encourage critical thinking about the non-neutral nature of data and explainability of machine learning [23].

A story-driven approach brings the human into the loop of machine learning and deep learning. Stories are essentially narratives about human experiences [60]. It is a way of persuasion and communication and a method of organizing information and constructing human experience in the world [60]. Using the NLP and knowledge graph technology, we can create interaction around storytelling between humans and machines so that human-in-the-loop becomes possible. In other words, stories make an interface between humans and machines for learning and problem-solving. Story based machine learning [97]

Our hypothesis of this study is that the understanding of the 311 call description can be converted into question/answering by designing a set of predictive models, such as 311 call type and working group classifying, time series prediction, and topic identification. We believe the traditional machine learning approach lacks efficient handling of the context in the corpus. Hence, a deep learning framework equipped with BERT, time series, topic modeling, and attention mechanism was designed to answer the 311 call questions.

The story of the machine learning pipeline has not been well studied in the literature. In this thesis, we hypothesis that the design of computational models for interactive plots capable of creating engaging stories and learn from experiences that are a meaningful interactive collaboration between machines and researchers (simulating interviewing)

at the end-to-end pipeline of AI and machine learning. The stories can be shared and learned from each other through the OCEL.AI process.

Furthermore, the human-machine partnership of the ML life cycle can be established to support open, networked collaboration and research. OCEL.AI has created a story-driven approach to teach data science/AI to computer science majors and non-computer science majors. A unique feature of the OCEL.AI storytelling approach is integrating communication and critical thinking education into data science education to make scientists “communicators.” Communication is among the most desirable skills in the job market. Simultaneously, storytelling also relates to domain disciplines that traditionally train communicators and storytellers, such as journalism and mass media. Storytelling bridges the cognitive gaps between disciplines.

OCEL.AI storytelling is not only serving the goals of data visualization but also theorizing and problem-solving. Data visualization can be described as a process of using information visualization techniques and narrative storytelling to achieve communication goals [27]. The purposes of data visualization are using data to inform, educate, advocate, influence, and persuade audiences. In contrast, the purpose of OCEL.AI storytelling is to serve the goals of data visualization and theorizing, and problem-solving. Ochs et al. pointed out that storytelling is a powerful tool for complex theory-building [69]. Axelrod and Kahn argued that storytelling goes hand-in-hand with modeling and can establish links between personal and social worlds via meaningful interpretation of large-scale datasets [4].

Following this school of pedagogical thoughts, our team designed five stages of

end-to-end story-driven data science/AI experiential learning, which entails data visualization. Our work is also influenced by science stories [33] and visualization ethics [23]. Science stories emphasize the importance of using scientists' discovery, rescue, and mystery stories to engage more people into problem-solving [33]. And, visualization ethics encourage critical thinking about the non-neutral nature of data and explainability of machine learning [23].

The OCEL.AI Storytelling framework integrates five components of data science education, which offers a well-rounded learning experience for undergraduates. It should be noted that the five components are not part of a linear process. Rather, they occur at various time points and sometimes simultaneously.

OCEL.AI has created a story-driven approach to teach data science/AI to computer science majors and non-computer science majors. Storytelling is the driving force of the entire process of OCEL.AI, from "Life" to "the Society." A unique feature of the OCEL.AI storytelling approach is it integrates communication and critical thinking education into data science education, to make scientists "communicators." Communication is among the most desirable skills in the job market. At the same time, storytelling also relates to domain disciplines that traditionally train communicators and storytellers, such as journalism and mass media. Storytelling bridges the cognitive gaps between disciplines.

Why are stories important? Include the two case studies? Stories are most important. If you are in advertising and journalism, you may want to ask whether a use case based upon just "my experience" is good enough for machines.

The answer could be Yes! It is likely that my story shares similarities with stories

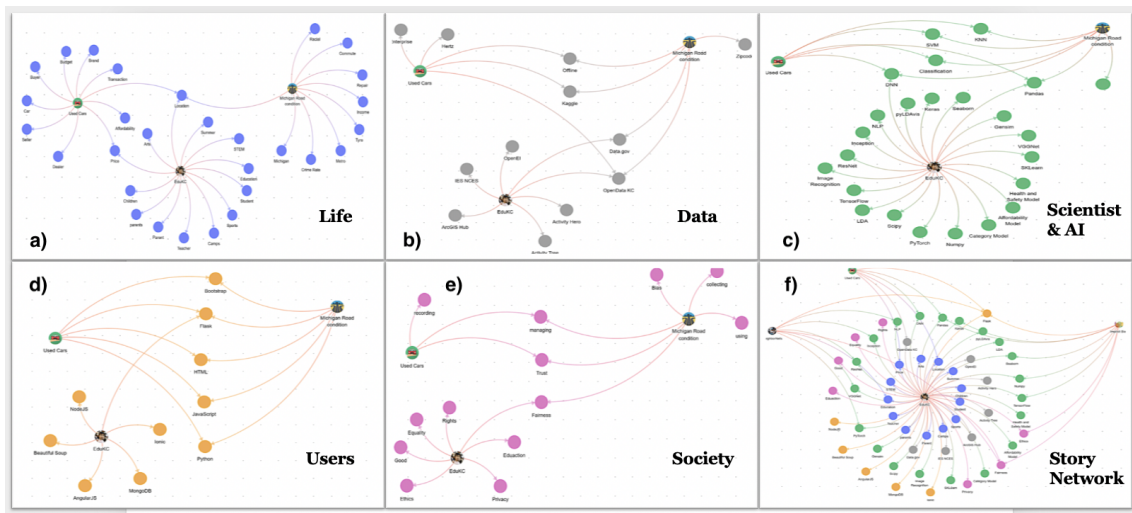


Figure 47: OCEL.AI StoryNet

of other people in the same social stratum. My story provides clues to machines to learn, particularly regarding key aspects, factors, and processes. As a result, the machine can use the frame of “my story” to read more stories from similar people, and acquire the pattern of the decision-making process, and thus make predictions/generalization.

The answer could be No! How about people who are outside of my social stratum? For example, I am living in the best school district. My accessibility to education resources is much greater than those in the other parts of the city, and I may not even care about inequality of education resources. It just does not matter that much to me!

In contrast, a single mom who works two jobs and lives in the urban center is desperate to find a safe, affordable, and educational preschool for her kids where such resources are sparse. “My story” is not her story! No worries. You simply need to do the same thing that we often do in audience research: talking to a single mom who fits that profile, and creating a “use case” based upon her experience.

This linear structure of the OCEL.AI story creates a story schema to generate a knowledge graph. This knowledge graph maps out the critical elements for each chapter using NLP technology. The visualization of all branches and their parts in an interactive map restores the non-linearity of data science problem-solving. The shared elements of different stories connect them to form a story network, OCEL.AI StoryNetwork.

This knowledge graph makes it possible for humans to interact with a dynamic knowledge base, get inspirations, learn technologies in context, identify relevant datasets, and deepen understandings of social and cultural implications.

The graphs are generated by identifying each story's most representative terms using LDA (Latent Dirichlet Allocation). The stories are connected through the common topics to help identify new insights, which may help create a new story. Figure 47(b) shows the data sources that were used for building models and apps for various stories. Figure 47(c) shows the model network, which identifies the relations between different stories through the lens of machine learning models and techniques used in them.

Such a network helps to identify the most suited models for a similar scenario based upon the use case. The standard methods used among different use cases help students select machine learning solutions intuitively based on the real-world impact rather than just learning the syntax. Figure 47(d) shows the relations among the applications of stories through the tools used to implement them. This graph also represents solutions to real-world problems. From this graph, the students can identify relevant and useful applications for each use case.

Figure 47(e) shows the network of concerns and causes related to the ethical considerations for each use case. These graphs add a layer of social and cultural depth to use cases and display a comprehensive picture of the solution in a broader social context. Figure 47(f) shows the five stories together as OCEL.AI StoryNetwork. A more comprehensive illustration of the same is given in Figure 46.

### **4.3 Case Study 3: CVE (Common Vulnerabilities and Exposures)**

The methodology and approach for analyzing CVE (Common Vulnerabilities and Exposures) data is same as what we have discussed for 311 cases and OCEL.AI in the previous sections. So, for the sake of simplicity and compactness will produce only the differences and results for the CVE case study here.

Data: The National Vulnerability Database is the U.S. government repository of standards-based vulnerability management data represented using the Security Content Automation Protocol (SCAP). It is a superset of the CVE® dictionary augmented with additional analysis, a database, and a fine-grained search engine. Usage restrictions of this resource are described in the NVD's FAQ:

All NVD data is freely available from our XML Data Feeds. There are no fees, licensing restrictions, or even a requirement to register. All NIST publications are available in the public domain according to Title 17 of the United States Code. Acknowledgment of the NVD when using our information is appreciated. In addition, please email [nvd@nist.gov](mailto:nvd@nist.gov) to let us know how the information is being used.

The screenshot shows the CVE Mitre website interface. At the top, there is a navigation bar with the CVE logo on the left and links for 'CVE List', 'CNAs', 'WGs', and 'Board'. On the right, there is an 'NVD' logo and links for 'Go to for: CVSS Scores' and 'CPE Info'. Below the navigation bar is a dark blue bar with white text for 'Search CVE List', 'Downloads', 'Data Feeds', 'Update a CVE Record', and 'Request CVE IDs'. A grey bar below that displays 'TOTAL CVE Records: 156053'. The main content area shows 'HOME > CVE > SEARCH RESULTS' and a 'Search Results' section. A message states 'There are 530 CVE Records that match your search.' Below this is a table with two columns: 'Name' and 'Description'. The table lists five CVE records with their IDs and brief descriptions of the vulnerabilities.

Name	Description
<a href="#">CVE-2021-3426</a>	There's a flaw in Python 3's pydoc. A local or adjacent attacker who discovers or is able to convince another lo and use it to disclose sensitive information belonging to the other user that they would not normally be able to. The flaw affects Python versions before 3.8.9, Python versions before 3.9.3 and Python versions before 3.10.0a7.
<a href="#">CVE-2021-33880</a>	The aaugustin websockets library before 9.1 for Python has an Observable Timing Discrepancy on servers who basic_auth_protocol_factory(credentials=...). An attacker may be able to guess a password via a timing attack.
<a href="#">CVE-2021-33571</a>	In Django 2.2 before 2.2.24, 3.x before 3.1.12, and 3.2 before 3.2.4, URLValidator, validate_ipv4_address, and octal literals. This may allow a bypass of access control that is based on IP addresses. (validate_ipv4_address).
<a href="#">CVE-2021-33509</a>	Plone through 5.2.4 allows remote authenticated managers to perform disk I/O via crafted keyword argument.
<a href="#">CVE-2021-33026</a>	The Flask-Caching extension through 1.10.1 for Flask relies on Pickle for serialization, which may lead to remote access to cache storage (e.g., filesystem, Memcached, Redis, etc.), they can construct a crafted payload, poison

Figure 48: CVE - Data

The Common Vulnerabilities and Exposures (CVE) system provides a reference-method for publicly known information-security vulnerabilities and exposures. The United States' National Cybersecurity FFRDC, operated by The Mitre Corporation, maintains the system, with funding from the US National Cyber Security Division of the US Department of Homeland Security. The system was officially launched for the public in September 1999.

The Security Content Automation Protocol uses CVE, and CVE IDs are listed on Mitre's system as well as in the US National Vulnerability Database. Figure 48 shows how the CVE database holds the records. The topic model for CVE is illustrated in Figure 49. The comparisons for performance on the test and train scores for CVE data are shown in Table 18.



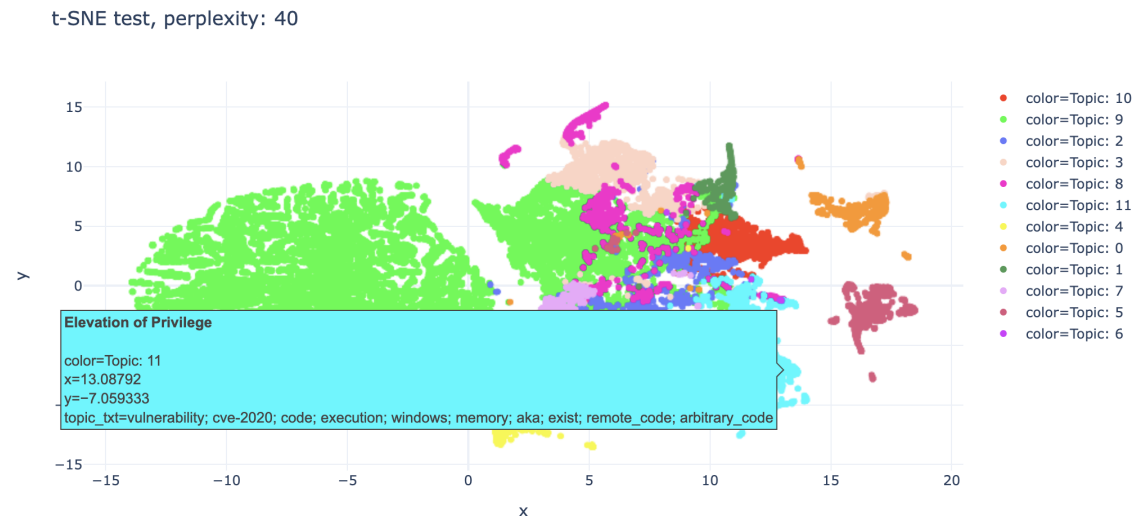


Figure 49: CVE - Topic Cluster

The following picture shows a cluster of the topics that are identified by the topic model and are represented using t-SNE format. As it can be observed, each topic has a collection of documents and each dot in the map corresponds to a document. One sample of such documents is shown in Figure 49 where the document that describes “Elevation of Privilege” as the title is highlighted. The corresponding keywords for the same topic can also be observed at the end of the information box.

The LDA analysis generates a distribution of the topic keywords and the corresponding principal component as shown in Figure 50. These keywords and principal components are visualized using library which generates an interactive visualization where the parameter  $\lambda$  can be modified and tested. The distance between the principal components shows how close / connected and how disconnected each topic is to the other topics.

As discussed in Chapter 3, it is a good idea to represent these topics in the form of word clouds. These word clouds help to visualize the topics in a more user-friendly

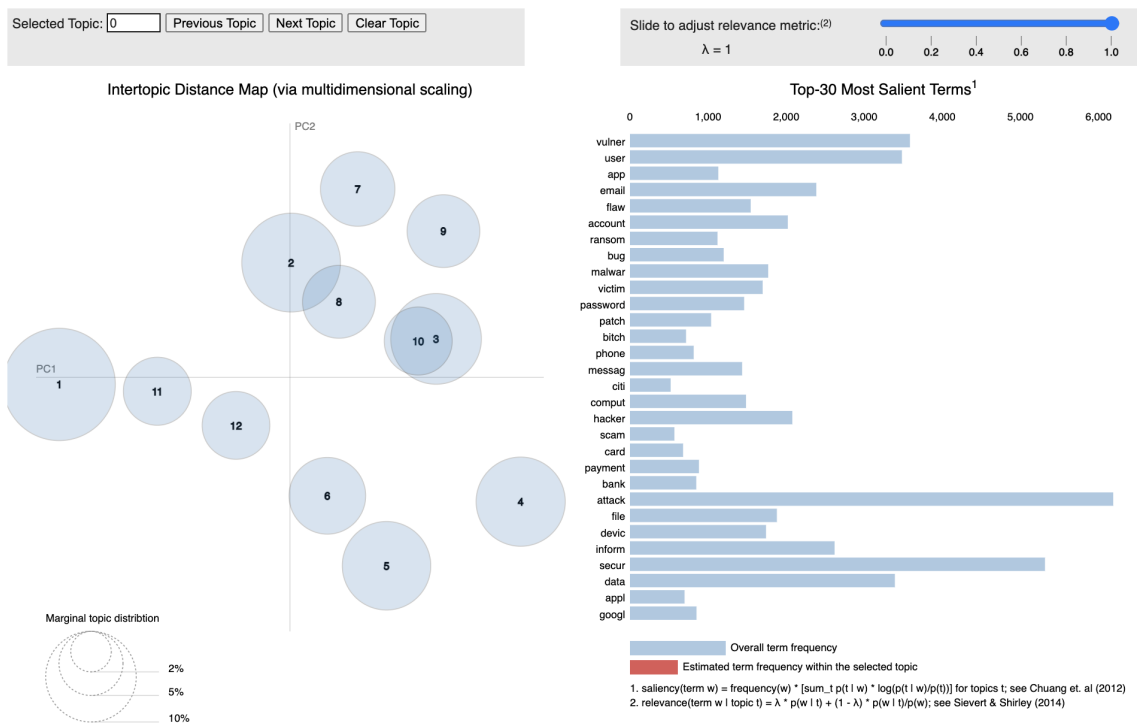


Figure 50: CVE - LDA Topics



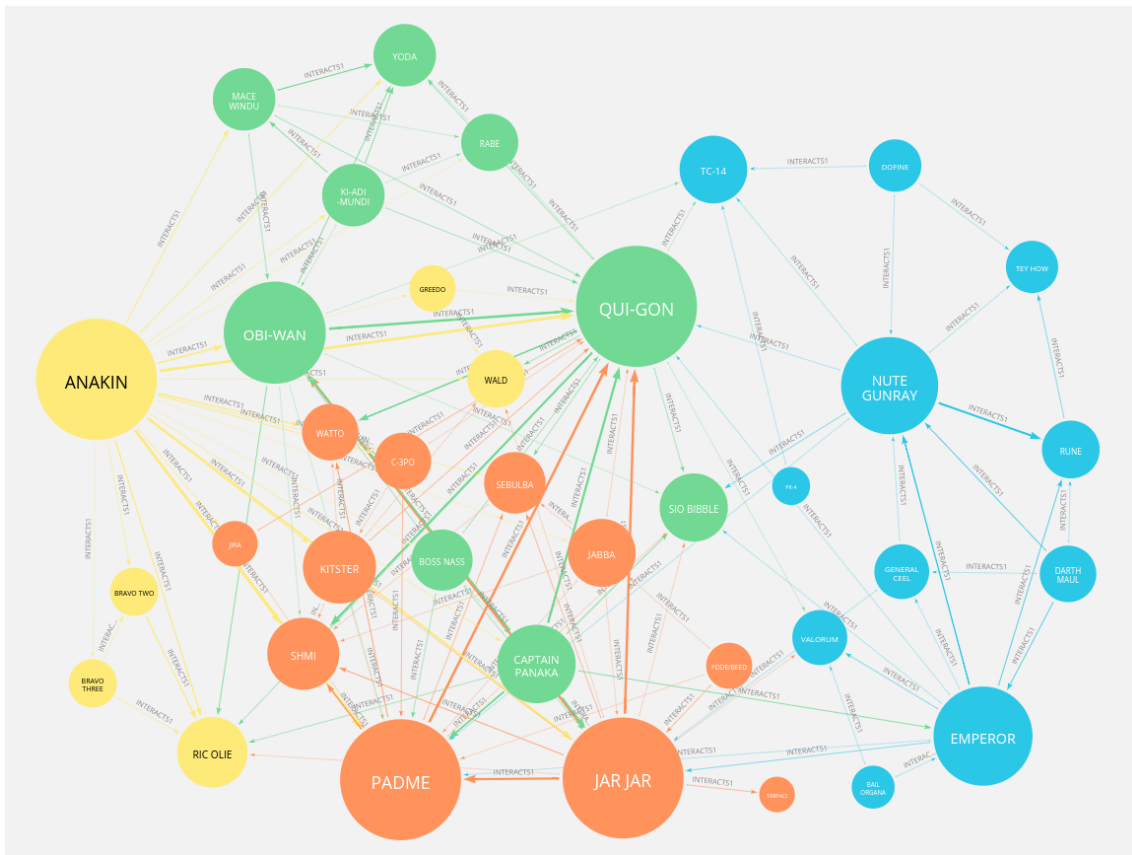


Figure 52: CVE - StoryNet

in Figure 52. Even though there are a large number of components since the application is interactive it is easier when we examine each story or component individually by grouping them together at different levels.

#### 4.4 Case Study 4: CASIE (CyberAttack Sensing and Information Extraction)

The methodology and approach for analyzing CASIE (CyberAttack Sensing and Information Extraction) data is same as what we have discussed for 311 cases and OCEL.AI in the previous sections. So, for the sake of simplicity and compactness will produce only

Model	Train FLOPs	train	test
BERT	5.4e20 (0.18x)	89.2	88
XLNet	3.6e21 (1.24x)	90.1	89.2
RoBERTa	3.2e21 (1.02x)	93.2	88.1
ALBERT	3.1e22 (10x)	92.3	89.9
StoryNet (Ours)	3.3e21 (1x)	93.7	90.2
AllenNLP	3.94e21 (1.27x)	91.4	84

Table 18: CVE Results

the differences and results for the CASIE case study here.

CASIE is a system that extracts information about cybersecurity events from text and populates a semantic model, with the ultimate goal of integration into a knowledge graph of cybersecurity data. It was trained on a new corpus of 1,000 English news articles from 2017–2019 that are labeled with rich, event-based annotations and that covers both cyberattack and vulnerability-related events.

Data: The corpus contains 1000 annotation and source files. This cybersecurity dataset [86] is focused on five event types: Databreach, Phishing, Ransom, Discover, and Patch. An example of a sample data is given in Figure 54 and also presented here for the sake of completeness:

```
<title>Ex-Chicago Public Schools worker accused of stealing info on 80,000 people in
latest data breach</title>
<source> https://chicago.suntimes.com/news/cps-data-breach-personal-information-former-
employee-chicago-public-schools/ </source>
<date> 2018_11_02 </date>
<text>
A former Chicago Public Schools worker faces several felony charges after
officials allege the worker stole personal information on about 80,000
employees, volunteers and vendors from a CPS database.
The former worker, Kristi Sims, was arrested Thursday; officers recovered the
stolen files after executing search warrants, according to CPS and Chicago
police officials. Sims, 28, is a former contractor who handled administrative
tasks for the Office of Safety and Security.
```

Sims was ordered released on her own recognizance at a bond hearing Friday at the Leighton Criminal Court Building by Judge Sophia Atcherson; Sims also was ordered not to access to the internet while the case continues.

In a letter to employees Thursday evening, CPS Chief Operating Officer Arnie Rivera said the district learned of the massive data breach Wednesday, the day after the information was stolen.

Among the data stolen were names, employee ID numbers, phone numbers, addresses, dates of birth, criminal arrest histories and DCFS findings. Social Security numbers were not taken, Rivera said.

"There was no indication that the information, which was in the individual's possession for approximately 24 hours, was used or disseminated to anyone in any way," Rivera added.

A CPS spokesman referred questions about the criminal charges to Chicago police, but Rivera said "CPS will work to ensure the individual is prosecuted to the fullest extent of the law."

CPD spokesman Anthony Guglielmi said Sims is also suspected of deleting the targeted files from the CPS database after they were stolen.

The digital equipment seized in the warrant is being analyzed, and a search warrant is underway for Sims's email account, Guglielmi said. Though police say they don't believe anyone other than Sims was in possession of the data, they hope to learn more about what might have been done with the information. This latest CPS data breach comes only a few months after the school district mistakenly sent a mass email that linked to the private information of thousands of students and families.

The email invited families to submit supplemental applications to selective enrollment schools. Attached at the bottom of the email was a link to a spreadsheet with the personal data of more than 3,700 students and families. In that incident, CPS apologized for the "unacceptable breach of both student information and your trust" and asked recipients of the email to delete the sensitive information. The data included children's names, home and cellphone numbers, email addresses and ID numbers.

</text>

The CASIE model defines five event subtypes along with their semantic roles and 20 event-relevant argument types (e.g., file, device, software, money). CASIE uses different deep neural networks approaches with attention and can incorporate rich linguistic features and word embeddings. The comparisons for performance on the test and train scores for CVE data are shown in Table 18.

```
▶ 1 payloads = ['Security is freedom from, or resilience against, potential harm (or other
2 'Security mostly refers to protection from hostile forces, but it has a wide range c
3 'The term is also used to refer to acts and systems whose purpose may be to provide
4 "The word 'secure' entered the English language in the 16th century. It is derived f
5 'A security referent is the focus of a security policy or discourse; for example, a
6 'Security referents may be persons or social groups, objects, institutions, ecosyste
7 'The security context is the relationships between a security referent and its Envir
8 'The means by which a referent provides for security (or is provided for) vary widel
9 'Coercive capabilities, including the capacity to project coercive power into the er
10 'Protective systems (e.g. lock, fence, wall, antivirus software, air defence system,
```

```
[ ] 1 for text in payloads:
2     payload = { "input_text": text }
3     output = qe.predict_boolq(payload)
4     pprint(output['Boolean Questions'])
```

```
['Is there such a thing as security?',
 'Is there a difference between true and false security?',
 'Is security the same as freedom from harm?']
['Is there such a thing as security?',
 'Is security the same as freedom from fear?',
 'Is there such thing as security?']
['Is there such thing as a security company?',
 'Is remote guarding the same as cyber security?',
 'Is there such thing as a cyber security system?']
['Is the word secure derived from the latin word?',
 'Is the word secure the same as freedom from anxiety?']
```

Figure 53: CASIE - Question Generation

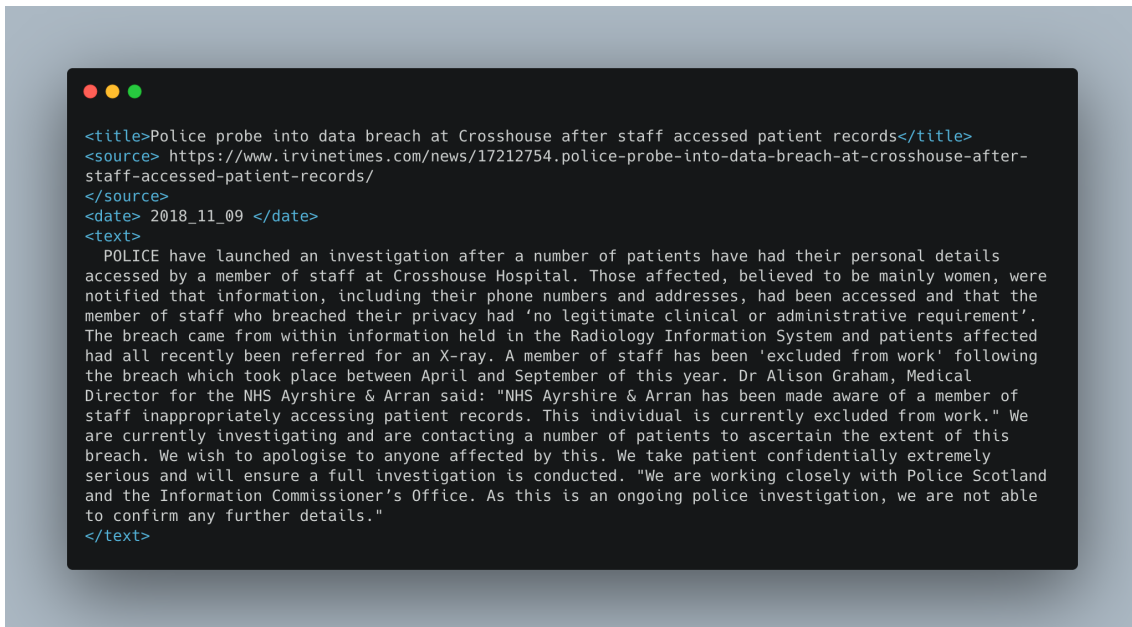


Figure 54: CASIE - Data

The annotated data is parsed as JSON. For annotating the data we made use of the automatic annotation as described in the model section of the previous chapter. After passing the data through the automatic annotation, each part of the annotation is marked in the JSON format. Example of the annotated data is shown here for your reference.

```

{
  ``content": ``A former Chicago Public Schools ... ``',
  ``sourcefile": ``10003.txt",
  ``cyberevent": {
    ``hopper": [
      {
        ``index": 0,
        ``relation": ``Same",
        ``events": [
          {
            ``index": ``E3",
            ``type": ``What",
            ``realis": ``Actual",
            ``nugget": {
              ``startOffset": 821,

```



```

    ``index": ``T11",
    ``endOffset": 832,
    ``text": ``data breach"
  },
  ``argument": [
    {
      ``index": ``T12",
      ``text": ``Wednesday",
      ``endOffset": 842,
      ``role": {
        ``type": ``Time"
      },
      ``startOffset": 833,
      ``type": ``When"
    }
  ],
  ``subtype": ``Databreach"
},
{
  ``index": ``E2",
  ``type": ``How",
  ``realis": ``Actual",
  ``nugget": {
    ``startOffset": 874,
    ``index": ``T8",
    ``endOffset": 884,
    ``text": ``was stolen"
  },
  ``subtype": ``Databreach"
},
{
  ``index": ``E1",
  ``type": ``Who",
  ``realis": ``Actual",
  ``nugget": {
    ``startOffset": 102,
    ``index": ``T1",
    ``endOffset": 107,
    ``text": ``stole"
  },
  ``argument": [
    {
      ``index": ``T7",
      ``text": ``the worker",

```

```

        ``endOffset": 101,
        ``role": {
            ``type": ``Attacker"
        },
        ``startOffset": 91,
        ``type": ``Person"
    },
    {
        ``index": ``T16",
        ``external_reference": {
            ``dbpediaURI": ``http://dbpedia.org/resource/Chicago_Public_Schools",
            ``wikidataid": ``Q2963340"
        },
        ``endOffset": 31,
        ``role": {
            ``type": ``Where"
        },
        ``text": ``Chicago Public Schools",
        ``startOffset": 9,
        ``type": ``Organization"
    }
],
``subtype": ``Databreach"
},
}
]
},
``info": {
    ``title": ``Ex-Chicago Public Schools worker accused of stealing info on 80,000 people in
    latest data breach",
    ``date": ``2018_11_02",
    ``type": ``text",
    ``link": ``https://chicago.suntimes.com/news/cps-data-breach-personal-information-
    former-employee-chicago-public-schools/"
}
}

```

We have presented the topics that we say found from the topic modeling on the CASIE data in figure. From the topics our cloud it is easy to figure out that each topic has a particular flavor. For example, topic one talks about email breach, information,



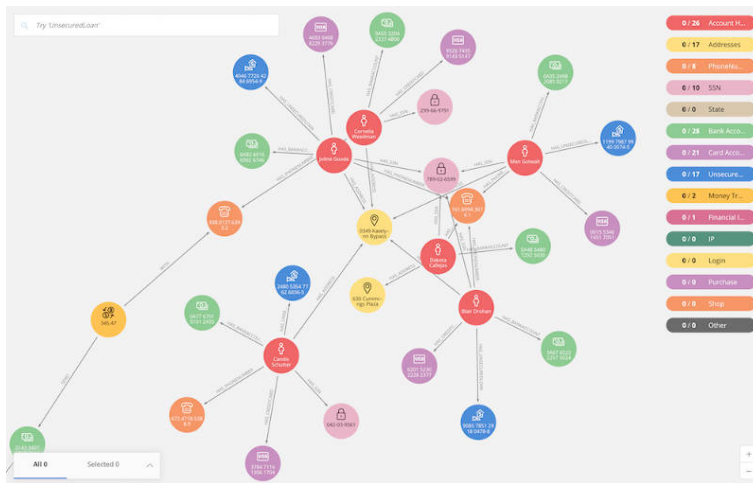


Figure 56: CASIE - StoryNet

Table 19: CASIE Results

Model	Train FLOPs	train	test
BERT	4.5e20 (0.16x)	90.2	89
XLNet	3.6e21 (1.12x)	92.1	89.3
RoBERTa	3.3e21 (1.03x)	93.2	89.9
ALBERT	3.1e22 (9.9x)	91.7	89.1
StoryNet (Ours)	3.2e21 (1x)	91.1	90.7
AllenNLP	3.72e21 (1.19x)	90.7	82

scam and probably anything that is related to the email accounts. Topic two show us the keywords related to security flaw, attack and vulnerability whereas, Topic 3 talks about user data, email accounts and so on. A cluster representation of these topics would look similar to the one that we shared before.

Let us examine the results for the CASIE data set. As presented in Table 19, our model outperforms other models as in the case with the location service that we have seen before. This shows that the model is rather very robust for different kinds of data sets.

Finally the outcome of the application of story net is presented in Figure 56. We

can observe that it is very easy to navigate between stories in this graph using the connections between the 5W1H components for a different cyber-security stories. Here different stories are color-coded using an algorithm in Neo4j.

## 4.5 Case Study Comparison

A consolidated comparison of the results for above case studies is shown in Figure 57. As you can see from the scores for each of the case studies it can be observed that the model performs very well on the 311 calls they reset. This is because we have more data available for this use case compared to all other use cases.

In the comparison we have finalized the outcomes from 5W1H layer for all the use cases. The data format for 311 calls is plain text, whereas the descriptions for other data sets have been extracted from either JSON for CVE XML format for CASIE. Since the data for OCEL.AI is very sparse.

The amount of data available for the CVE data set and the CASIE data set are almost on a similar order. So the scores for those two data sets also fall into similar ranges. The outcome for CASIE StoryNet is illustrated in Figure 56 and the corresponding word cloud for the topic modeling analysis is presented in Figure 55.

### 4.5.1 Tech Stack

The application in the backend is set up using Java Vault API, NodeJS server and Neo4j graph database. The front-end of the application is made using the angular framework combined with HTML 5, CSS3, JavaScript and D3 JS library for graph generation. The architecture for the combined backend and front end is shown in Figure 58. If you

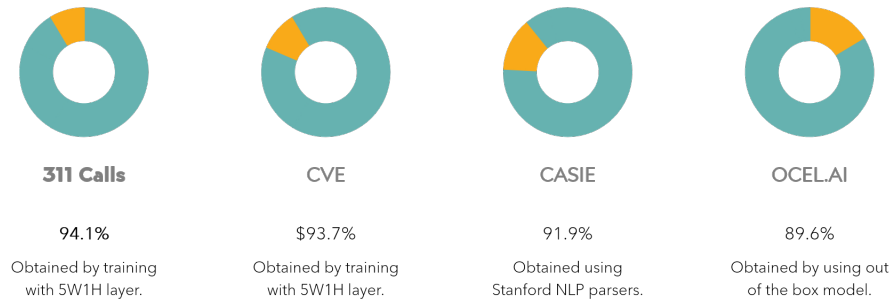


Figure 57: Performance Comparison - Use Cases

Table 20: Technologies Used in Apps

MongoDB	A NoSQL database for storing and querying un-structured data.
NodeJS	A javascript runtime environment to run a back-end server for an application.
Beautiful Soup	A python library useful to parse and extract data from HTML (web scraping).
Flask	A lightweight python web server useful to create APIs.
AngularJS	A front-end web framework used to create single page applications.
Ionic	A javascript framework for developing hybrid mobile applications with Javascript

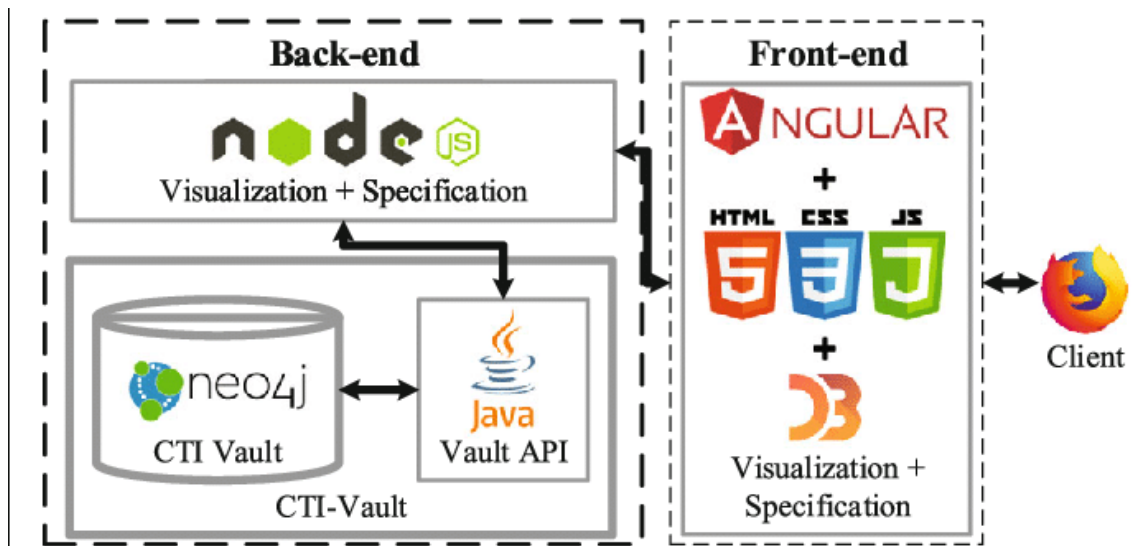


Figure 58: Application Architecture

other levels which are used as a part of the application are described in Table 20.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

#### 5.1 Conclusion

News articles are considered as one of the top sources for providing information on ground backing news, discussions on trending topics, or topic of interests to the users. The discussions on News articles could be the references to the occurrence of potential events which could be significant to the users. However, detecting events from News articles is a difficult task due to the nature of tweets.

News articles are short in length, often noisy, and users often do not follow grammatical structure while posting on News articles. In this thesis, we presented a News articles event detection system by segmenting the tweets with the 5W1H components, which are essential semantic constituents of information. We performed experiments with our 5W1H based segmentation of the tweets and compared them with a simple TF-IDF vectorization approach.

We used the recent state-of-the-art technology for generating contextualized embeddings from the 5W1H components, which we later used for generating the clusters. The dataset used in our work are very generic, and the approach adopted could be extended to detect specific events related to disasters, sports and terror activities. In future, we intend to include experiments with other clustering techniques and compare the clustering quality. We also intend to test our approach on accessible event detection datasets



and compare our approach with other similar event detection systems.

## 5.2 Future Work

Future studies can conduct a comparative evaluation with other frameworks using other data from various sources as future work. Built upon this framework, researchers can enhance the novelty of the methodology for creating the open community framework and interpret the empirical findings through empirical measurement and assessment. This framework will be extended to improve its handling of free text and images in the long run. These works would make it possible to address the ultimate questions for sustainable communities.

We should further explore this story-driven interface and advance the story network in other both custom and generic domains. Education and psychological research are also needed to study computer science students' and teachers' perception, acceptance, and self-efficacy of incorporating storytelling in data science education.

### 5.2.1 Cause and Consequence prediction:

Causal: estimate  $P(Y||X)$ , when  $X \Rightarrow Y$  (predict effect from cause); or Anti-causal: estimate  $P(Y||X)$ , when  $Y \Rightarrow X$  (predict cause from effect). The answer is crucial to all further causal analyses of the problem. It substantially impacts the applicability of decision supporting in conducting community service according to the community's particular context. Furthermore, we will explore which model, either generative or discriminative, would be more preferred to confirm the process's effectiveness.

For example, we can ask questions such as *whether can a cause of worse conditions can be described by frequent calls for potholes, dirty sidewalks, downed trees, graffiti, noise? Whether can worse conditions be a cause of increased or decreased propensity?* The volume and actual geographic distributions of 311 calls can be used to identify the levels of citizen engagement, local civic engagement, and the quality of government services to neighborhoods [61]. An overview of the process of question generation which is an extension to finding the 5W1H component is shown in Figure 53.

#### 5.2.2 Missing data prediction:

We've currently utilized the answers to 5W1H components which are identified by the models or humans. But, with the addition of a predictive model it is possible to fill in the missing data for any of the answers that do we not have data about.

#### 5.2.3 Metaverse extension:

Further research can be done to extend this framework into the virtual reality domain by making use of a few models a place in Facebook AI research (FAIR). It was all world consists of sequence of stories. Generating and making use of the existing stories, virtual world can be generated and we should be able to map the existing Omniverse models to the story to create the world. It will be really interesting to watch the stories come to life in the Omniverse.

## REFERENCE LIST

- [1] Abdi, A., Idris, N., and Ahmad, Z. QAPD: an ontology-based question answering system in the physics domain. *Soft Computing* 22, 1 (2018), 213–230.
- [2] Ammanabrolu, P., Cheung, W., Tu, D., Broniec, W., and Riedl, M. O. Bringing stories alive: Generating interactive fiction worlds. *arXiv preprint arXiv:2001.10161* (2020).
- [3] Ansari, A., Maknojia, M., and Shaikh, A. Intelligent question answering system based on artificial neural network. In *2016 IEEE International Conference on Engineering and Technology (ICETECH)* (2016), IEEE, pp. 758–763.
- [4] Axelrod, D. B., and Kahn, J. Intergenerational family storytelling and modeling with large-scale data sets. In *Proceedings of the 18th ACM International Conference on Interaction Design and Children* (2019), pp. 352–360.
- [5] Baker, C. F., Fillmore, C. J., and Lowe, J. B. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1* (1998), pp. 86–90.
- [6] Bakshi, K. Considerations for big data: Architecture and approach. In *2012 IEEE aerospace conference* (2012), IEEE, pp. 1–7.

- [7] Bar-Yam, Y. Complexity rising: From human beings to human civilization, a complexity profile, 2000.
- [8] Barley, W. C., Treem, J. W., and Leonardi, P. M. Experts at coordination: Examining the performance, production, and value of process expertise. *Journal of Communication* 70, 1 (2020), 60–89.
- [9] Bhoir, V., and Potey, M. Question answering system: A heuristic approach. In *The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014)* (2014), IEEE, pp. 165–170.
- [10] Blei, D. M., Ng, A. Y., and Jordan, M. I. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.
- [11] Borovykh, A., Bohte, S., and Oosterlee, C. W. Conditional time series forecasting with convolutional neural networks. *arXiv preprint arXiv:1703.04691* (2017).
- [12] Bowman, S. R., Angeli, G., Potts, C., and Manning, C. D. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing* (Lisbon, Portugal, Sept. 2015), Association for Computational Linguistics, pp. 632–642.
- [13] Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165* (2020).

- [14] Chakma, K., and Das, A. A 5w1h based annotation scheme for semantic role labeling of english tweets. *Computación y Sistemas* 22, 3 (2018), 747–755.
- [15] Chali, Y., Hasan, S. A., and Joty, S. R. Improving graph-based random walks for complex question answering using syntactic, shallow semantic and extended string subsequence kernels. *Information Processing & Management* 47, 6 (2011), 843–855.
- [16] Chan, Y.-H., and Fan, Y.-C. A Recurrent BERT-based Model for Question Generation. In *Proceedings of the 2nd Workshop on Machine Reading for Question Answering* (2019), pp. 154–162.
- [17] Chen, Q., Zhuo, Z., and Wang, W. Bert for joint intent classification and slot filling. *arXiv preprint arXiv:1902.10909* (2019).
- [18] Chen, W., Matusov, E., Khadivi, S., and Peter, J.-T. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628* (2016).
- [19] Chen, Z., Zhang, H., Zhang, X., and Zhao, L. Quora question pairs. *URL <https://www.kaggle.com/c/quora-question-pairs>* (2018).
- [20] Chimmula, V. K. R., and Zhang, L. Time series forecasting of COVID-19 transmission in Canada using LSTM networks. *Chaos, Solitons & Fractals* 135 (2020), 109864.

- [21] Clark, K., Luong, M.-T., Le, Q. V., and Manning, C. D. Electra: Pre-training text encoders as discriminators rather than generators. *arXiv preprint arXiv:2003.10555* (2020).
- [22] Collins, H. M., and Evans, R. The third wave of science studies: Studies of expertise and experience. *Social studies of science* 32, 2 (2002), 235–296.
- [23] Correll, M. Ethical dimensions of visualization research. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems* (2019), pp. 1–13.
- [24] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018).
- [25] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)* (Minneapolis, Minnesota, June 2019), Association for Computational Linguistics, pp. 4171–4186.
- [26] Dwivedi, S. K., and Singh, V. Research and reviews in question answering system. *Procedia Technology* 10 (2013), 417–424.
- [27] Echeverria, V., Martinez-Maldonado, R., Granda, R., Chiluiza, K., Conati, C., and Shum, S. B. Driving data storytelling from learning design. In *Proceedings of the*

- 8th international conference on learning analytics and knowledge* (2018), pp. 131–140.
- [28] Erlingsson, C., and Brysiewicz, P. A hands-on guide to doing content analysis. *African Journal of Emergency Medicine* 7, 3 (2017), 93–99.
- [29] Fernando, S., and Stevenson, M. A semantic similarity approach to paraphrase detection. In *Proceedings of the 11th annual research colloquium of the UK special interest group for computational linguistics* (2008), Citeseer, pp. 45–52.
- [30] Ferrucci, D. A. Introduction to “this is watson”. *IBM Journal of Research and Development* 56, 3.4 (2012), 1–1.
- [31] Fletcher-Watson, S., Adams, J., Brook, K., Charman, T., Crane, L., Cusack, J., Leekam, S., Milton, D., Parr, J. R., and Pellicano, E. Making the future together: Shaping autism research through meaningful participation. *Autism* 23, 4 (2019), 943–953.
- [32] Ghosh, S., Vinyals, O., Strope, B., Roy, S., Dean, T., and Heck, L. Contextual lstm (clstm) models for large scale nlp tasks. *arXiv preprint arXiv:1602.06291* (2016).
- [33] Green, S. J., Grorud-Colvert, K., and Mannix, H. Uniting science and stories: perspectives on the value of storytelling for communicating science, 2018.
- [34] Griffin, P. F. The correlation of english and journalism. *The English Journal* 38, 4 (1949), 189–194.

- [35] Guo, L., Lei, Y., Xing, S., Yan, T., and Li, N. Deep convolutional transfer learning network: A new method for intelligent fault diagnosis of machines with unlabeled data. *IEEE Transactions on Industrial Electronics* 66, 9 (2018), 7316–7325.
- [36] Hasan, M., Orgun, M. A., and Schwitter, R. Real-time event detection from the Twitter data stream using the TwitterNews+ Framework. *Information Processing & Management* 56, 3 (2019), 1146–1165.
- [37] He, L., Lewis, M., and Zettlemoyer, L. Question-answer driven semantic role labeling: Using natural language to annotate natural language. In *Proceedings of the 2015 conference on empirical methods in natural language processing* (2015), pp. 643–653.
- [38] Hendrycks, D., and Gimpel, K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *ArXiv* (2016).
- [39] Hewamalage, H., Bergmeir, C., and Bandara, K. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting* 37, 1 (2021), 388–427.
- [40] Hu, Y., Zhai, K., Eidelman, V., and Boyd-Graber, J. Polylingual tree-based topic models for translation domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2014), pp. 1166–1176.



- [41] Jameson, A., Berendt, B., Gabrielli, S., Cena, F., Gena, C., Venero, F., Reinecke, K., et al. Choice architecture for human-computer interaction. *ArXiv* (2014).
- [42] Joshi, M., Choi, E., Weld, D. S., and Zettlemoyer, L. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (2017), pp. 1601–1611.
- [43] kcmo. KCMO 311. <https://www.kcmo.gov/city-hall/departments/neighborhoods-housing-services>, 2013. [Online; accessed 13-September-2020].
- [44] Keele, S., et al. Guidelines for performing systematic literature reviews in software engineering. Tech. rep., Citeseer, 2007.
- [45] Kwiatkowski, T., Palomaki, J., Redfield, O., Collins, M., Parikh, A., Alberti, C., Epstein, D., Polosukhin, I., Devlin, J., Lee, K., et al. Natural Questions: A Benchmark for Question Answering Research, 2019.
- [46] Lai, V. D., Dernoncourt, F., and Nguyen, T. H. Extensively matching for few-shot learning event detection. *arXiv preprint arXiv:2006.10093* (2020).
- [47] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. Albert: A lite bert for self-supervised learning of language representations. *arXiv preprint arXiv:1909.11942* (2019).

- [48] Lan, Z., Chen, M., Goodman, S., Gimpel, K., Sharma, P., and Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations, 2020.
- [49] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36, 4 (2020), 1234–1240.
- [50] Lende, S. P., and Raghuwanshi, M. Question answering system on education acts using NLP techniques. In *2016 world conference on futuristic trends in research and innovation for social welfare (Startup Conclave)* (2016), IEEE, pp. 1–6.
- [51] Li, C., Sun, A., and Datta, A. Twevent: segment-based event detection from tweets. In *Proceedings of the 21st ACM international conference on Information and knowledge management* (2012), pp. 155–164.
- [52] Li, S., Zhang, Y., and Pan, R. Bi-directional recurrent attentional topic model. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 6 (2020), 1–30.
- [53] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692* (2019).
- [54] Livieris, I. E., Pintelas, E., and Pintelas, P. A CNN–LSTM model for gold price time-series forecasting. *Neural computing and applications* 32, 23 (2020), 17351–17360.

- [55] Malik, N., Sharan, A., and Biswas, P. Domain knowledge enriched framework for restricted domain question answering system. In *2013 IEEE International Conference on Computational Intelligence and Computing Research* (2013), IEEE, pp. 1–7.
- [56] Maxime. What is a Transformer. <https://medium.com/inside-machine-learning/what-is-a-transformer-d07dd1fbec04>, 2019. [Online].
- [57] McBride, B. The resource description framework (RDF) and its vocabulary description language RDFS. In *Handbook on ontologies*. Springer, 2004, pp. 51–65.
- [58] Medsker, L. R., and Jain, L. Recurrent neural networks. *Design and Applications* 5 (2001), 64–67.
- [59] Mihaylov, T., and Nakov, P. SemanticZ at SemEval-2016 Task 3: Ranking relevant answers in community question answering using semantic similarity based on fine-tuned word embeddings. *arXiv preprint arXiv:1911.08743* (2019).
- [60] Miller-Day, M., and Hecht, M. L. Narrative means to preventative ends: A narrative engagement framework for designing prevention interventions. *Health communication* 28, 7 (2013), 657–670.
- [61] Minkoff, S. L. NYC 311: A tract-level analysis of citizen–government contacting in New York City. *Urban Affairs Review* 52, 2 (2016), 211–246.

- [62] Morabia, K., Murthy, N. L. B., Malapati, A., and Samant, S. SEDTWik: segmentation-based event detection from tweets using Wikipedia. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop* (2019), pp. 77–85.
- [63] Nangia, N., Williams, A., Lazaridou, A., and Bowman, S. R. The repeal 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172* (2017).
- [64] Narayan, S., Cohen, S. B., and Lapata, M. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745* (2018).
- [65] Neil, D., Pfeiffer, M., and Liu, S.-C. Phased lstm: Accelerating recurrent network training for long or event-based sequences. In *Advances in neural information processing systems* (2016), pp. 3882–3890.
- [66] Ngo, N. T., Nguyen, T. N., and Nguyen, T. H. Learning to select important context words for event detection. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining* (2020), Springer, pp. 756–768.
- [67] Niu, T., Wang, J., Lu, H., Yang, W., and Du, P. Developing a deep learning framework with two-stage feature selection for multivariate financial time series forecasting. *Expert Systems with Applications* 148 (2020), 113237.

- [68] Norambuena, B., Horning, M., and Mitra, T. Evaluating the inverted pyramid structure through automatic 5wh extraction and summarization. In *Computational Journalism Symposium* (2020).
- [69] Ochs, E., Taylor, C., Rudolph, D., and Smith, R. Storytelling as a theory-building activity. *Discourse processes* 15, 1 (1992), 37–72.
- [70] Okoli, C., and Schabram, K. A guide to conducting a systematic literature review of information systems research. *ArXiv* (2010).
- [71] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).
- [72] Palmer, M. D. Gildea e P. Kingsbury, The Proposition Bank: A Corpus Annotated with Semantic Roles. *Computational Linguistics Journal* 31, 1 (2005).
- [73] Peinelt, N., Nguyen, D., and Liakata, M. tBERT: Topic models and BERT joining forces for semantic similarity detection. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (2020), pp. 7047–7055.
- [74] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365* (2018).

- [75] Pinto, D., Gómez-Adorno, H., Vilarino, D., and Singh, V. K. A graph-based multi-level linguistic representation for document understanding. *Pattern recognition letters* 41 (2014), 93–102.
- [76] Pudaruth, S., Boodhoo, K., and Goolbudun, L. An intelligent question answering system for ict. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (2016), IEEE, pp. 2895–2899.
- [77] Qin, Z., Thint, M., and Huang, Z. Ranking answers by hierarchical topic models. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (2009), Springer, pp. 103–112.
- [78] Radford, A., Narasimhan, K., Salimans, T., and Sutskever, I. Improving language understanding by generative pre-training, 2018.
- [79] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. *OpenAI blog* 1, 8 (2019), 9.
- [80] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer, 2019.
- [81] Rajpurkar, Zhang, Lopyrev, and Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv preprint arXiv:1606.05250v3* (2016).
- [82] Roberts, A., Raffel, C., and Shazeer, N. How Much Knowledge Can You Pack Into the Parameters of a Language Model? *arXiv preprint arXiv:2002.08910* (2020).

- [83] Rogers, B., Bertram, N., Gersonius, B., Gunn, A., Löwe, R., Murphy, C., Pashman, R., Radhakrishnan, M., Urich, C., Wong, T., et al. An interdisciplinary and catchment approach to enhancing urban flood resilience: a Melbourne case. *Philosophical Transactions of the Royal Society A* 378, 2168 (2020), 20190201.
- [84] Sandyvarma et al., S. A. Covid-19: BERT + MeSH Enabled Knowledge Graph. <https://www.kaggle.com/sandyvarma/covid-19-bert-mesh-enabled-knowledge-graph>, April 2020.
- [85] Sanh, V., Debut, L., Chaumond, J., and Wolf, T. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, 2019.
- [86] Satyapanich, T., Ferraro, F., and Finin, T. Casie: Extracting cybersecurity event information from text. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2020), vol. 34, pp. 8749–8757.
- [87] Schneider, H., Eiband, M., Ullrich, D., and Butz, A. Empowerment in HCI-A survey and framework. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (2018), pp. 1–14.
- [88] Schuler, K. K. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania, 2005.
- [89] Seth, Y. BERT Explained. <https://yashueth.blog/2019/06/12/bert-explained-faqs-understand-bert-working/>, 2019. [Online].

- [90] Sherstinsky, A. Fundamentals of recurrent neural network (RNN) and long short-term memory (LSTM) network. *Physica D: Nonlinear Phenomena* 404 (2020), 132306.
- [91] Shi, P., and Lin, J. Simple BERT Models for Relation Extraction and Semantic Role Labeling. arXiv 2019. *arXiv preprint arXiv:1904.05255* (1904).
- [92] Sirignano, J. A. Deep learning for limit order books. *Quantitative Finance* 19, 4 (2019), 549–570.
- [93] Smyl, S. A hybrid method of exponential smoothing and recurrent neural networks for time series forecasting. *International Journal of Forecasting* 36, 1 (2020), 75–85.
- [94] Stoll, J. Number of commercial TV stations in the U.S. 1950-2017, 2021.
- [95] Stveshawn. Contextual topic identification for Steam reviews. [https://github.com/Stveshawn/contextual\\_topic\\_identification](https://github.com/Stveshawn/contextual_topic_identification)(2019).
- [96] Tang, J., Meng, Z., Nguyen, X., Mei, Q., and Zhang, M. Understanding the limiting factors of topic modeling via posterior contraction analysis. In *International Conference on Machine Learning* (2014), PMLR, pp. 190–198.
- [97] Tangherlini, T. R., Roychowdhury, V., Glenn, B., Crespi, C. M., Bandari, R., Wadia, A., Falahi, M., Ebrahimzadeh, E., and Bastani, R. “Mommy Blogs” and the vaccination exemption narrative: results from a machine-learning approach for



- story aggregation on parenting social media sites. *JMIR public health and surveillance* 2, 2 (2016), e166.
- [98] Taylor, S. J., and Letham, B. Forecasting at scale. *The American Statistician* 72, 1 (2018), 37–45.
- [99] Tran, Q. H., Tran, D.-V., Vu, T., Le Nguyen, M., and Pham, S. B. JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)* (2015), pp. 215–219.
- [100] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems* (2017), pp. 5998–6008.
- [101] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is All you Need. In *Advances in Neural Information Processing Systems* (2017), I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, Curran Associates, Inc.
- [102] Venkataram, H. S., Mattmann, C. A., and Penberthy, S. TopiQAL: Topic-aware Question Answering using Scalable Domain-specific Supercomputers. In *2020 IEEE/ACM Fourth Workshop on Deep Learning on Supercomputers (DLS)* (2020), IEEE, pp. 48–55.

- [103] Voorhees, E. M., and Tice, D. M. Building a question answering test collection. In *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval* (2000), pp. 200–207.
- [104] Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., and Bowman, S. R. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461* (2018).
- [105] Wang, C., Akbik, A., Chiticariu, L., Li, Y., Xia, F., and Xu, A. CROWD-IN-THE-LOOP: A hybrid approach for annotating semantic roles. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), pp. 1913–1922.
- [106] Wang, L., Yao, J., Tao, Y., Zhong, L., Liu, W., and Du, Q. A reinforced topic-aware convolutional sequence-to-sequence model for abstractive text summarization. *arXiv preprint arXiv:1805.03616* (2018).
- [107] Wang, W., Auer, J., Parasuraman, R., Zubarev, I., Brandyberry, D., and Harper, M. A question answering system developed as a project in a natural language processing course. In *ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems* (2000).
- [108] Warstadt, A., Singh, A., and Bowman, S. R. Cola: The corpus of linguistic acceptability (with added annotations). *ArXiv* (2019).
- [109] White, K. Publications Output: U.S. Trends and International Comparisons, 2019.

- [110] Williams, A., Nangia, N., and Bowman, S. A Broad-Coverage Challenge Corpus for Sentence Understanding through Inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)* (2018), Association for Computational Linguistics, pp. 1112–1122.
- [111] Wu, W.-N. Determinants of Citizen-Generated Data in a Smart City: Analysis of Open 311 User Behavior. *Sustainable Cities and Society* (2020), 102167.
- [112] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K. Convolutional neural networks: an overview and application in radiology. *Insights into imaging* 9, 4 (2018), 611–629.
- [113] Yang, P., Fang, H., and Lin, J. Anserini: Reproducible ranking baselines using Lucene. *Journal of Data and Information Quality (JDIQ)* 10, 4 (2018), 1–20.
- [114] Yang, Y., Yuan, S., Cer, D., Kong, S.-y., Constant, N., Pilar, P., Ge, H., Sung, Y.-H., Strope, B., and Kurzweil, R. Learning semantic textual similarity from conversations. *arXiv preprint arXiv:1804.07754* (2018).
- [115] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems* (2019), pp. 5753–5763.

- [116] Yu, H., and Cao, Y.-g. Automatically extracting information needs from ad hoc clinical questions. In *AMIA annual symposium proceedings (2008)*, vol. 2008, American Medical Informatics Association, p. 96.
- [117] Zhang, Z., Zhang, Z., Chen, H., and Zhang, Z. A joint learning framework with bert for spoken language understanding. *IEEE Access* 7 (2019), 168849–168858.
- [118] Zhao, Z., Sun, J., Mao, Z., Feng, S., and Bao, Y. Determining the topic hashtags for chinese microblogs based on 5W model. In *International Conference on Big Data Computing and Communications* (2016), Springer, pp. 55–67.

## VITA

Srichakradhar Reddy Nagireddy is a Master's student at UMKC. He worked as a Software Developer at Accenture, Full Stack Developer at a start-up SetuServ and as a Technical Lead at Tata Consultancy services before he joined UMKC for MSCS in Spring 2020. Being an active participant and winner of 3 Hack-A-Roos (UMKC conducted hackathos), he also worked as a Research Assistant under Dr. Yugyung Lee since Summer of 2020 contributing to 3 different projects - OCEL.AI (Open Collaborative Experiential Learning), NSF S&CC (Smart & Connected Communities) and Cybersecurity in Minecraft. He has 3 publications in IJAER, JAP, AMR, 2 publications under review and 1 conference paper at AEJMC. His research interests mainly include Semantic Web, Multimodal Representations, Question Answering and Language Models. He is currently working as Software Engineer at Google LLC in Mountain View.