CS4984 / CS5984 Big Data Text Summarization - Team 15 Report Maryland Shooting

Bipasha Banerjee
Prapti Khawas
Shuqi Zhao
Yiyang Fan
Yoonjin Kim

Instructor: Dr. Edward A. Fox

Department of Computer Science
Virginia Tech
Blacksburg, VA 24061
December 15, 2018

Contents

List of Figures

List of Tables

1 Executive Summary

The goal of this work is to generate summaries of two Maryland shooting events from a large collection of web pages related to a shooting at Great Mills High School and another at the Capital Gazette newsroom. Since our team did not have prior experience with Computational Linguistics / Natural Language Processing (NLP), we followed an approach where we built summaries using 10 different methods, as suggested by course instructor Dr. Edward Fox, with each method being more sophisticated than the previous ones, to enable learning of key concepts in NLP.

First, we started with finding a set of most frequent important words. Then, we found other words occurring in the articles which mean the same as the frequent words found. Along with the synonyms, we found sets of hypernyms and hyponyms. We identified a set of words constrained by POS, e.g., nouns and verbs. We then tried out various classification techniques in Apache Mahout to classify the documents into the two different events and eliminate irrelevant documents. Next, we identified a set of frequent and important named entities using NLTK and SpaCy Named Entity Recognition (NER) modules. We identified a set of important topics identified using Latent Dirichlet Allocation (LDA). We then generated clusters of documents using K-means. Next, we extracted a set of values for each slot matching collection semantics using regular expressions and generated a readable summary explaining the slots and values using a Context Free Grammar we developed. Finally, we used the Pointer Generator deep learning approach to generate a readable abstractive summary.

Using the above approach, we generated two extractive summaries for newsroom shooting event and school shooting event with ROUGE-1 scores around 0.33 and 0.26 respectively. For the abstractive summaries, that we generated, the ROUGE-1 score was 0.36 for newsroom shooting event and 0.20 for school shooting event. We also evaluated the summaries at sentence level and we found that the abstractive school shooting summary had a higher ROUGE-1 score, being 0.88, than abstractive newsroom shooting summary with 0.73.

We employed the Hadoop MapReduce framework to speed up the processing time for our large collection. We used various other tools like the NLTK language processing library and Apache Mahout, a distributed linear algebra framework to simplify our development. We learned that a variety of different methods and techniques which suit the collection are necessary in order to provide an accurate summary. Since the collection contained a copious amount of irrelevant information, filtering them was a complicated yet necessary step.

2 Introduction

The purpose of the project was to summarize a big collection of data. The given corpus contained articles related to Maryland shooting. We later noticed that two different shooting events were present in our collection of data. One of them was the unfortunate shooting at Maryland's Capital Gazette Newsroom. The other event was the shooting at Great Mills High School in Maryland. Hence, our primary goal was to produce summaries of both the incidents using various text summarization techniques.

2.1 Background of the Dataset

Initially, we were assigned the Tunisia Election dataset. However, we were later assigned Maryland Shooting as the dataset we would use for the summarization technique. When we began with the first task, we realized that the collection contained two events related to Maryland shooting - one being at the Capital Gazette (a.k.a. Maryland Newsroom shooting) and the other being at Great Mills High School (a.k.a. Maryland School shooting). We decided to segregate the articles based on the presence of keywords - "Capital Gazette" for the newsroom shooting and "Great Mill" for the school shooting. Later on, we tried various supervised classification techniques as described in Section 4.4.

2.2 Pre-Processing the Data

We carried out the following steps in order to pre-process the dataset:

1. Processed WARC and CDX files (initial dataset) on DLRL cluster by cleaning the content using OpenNLP's sentence detector tool.
2. Converted cleaned content into JSON format and copied the resulting JSON file to our directory.
3. Indexed the JSON file on Solr.
4. Converted JSON file into text files each containing content from an article.
5. Cleaned the articles by eliminating sentences containing unwanted words such as 'login', 'advertisement', 'video', etc.
6. Segregated articles into two groups using keywords "Capital Gazette" and "Great Mills".

3 Literature Review

The resources provided by the instructor in Canvas were very useful. Especially, in the first few modules, when the team members were relatively inexperienced, we received help from the unit-wise guide provided in Canvas. It has detailed steps that we could follow to achieve our desired goal. A paper (1) by Allahyari et al. discusses all the text summarization techniques developed so far and describes the effectiveness and shortcomings of each.

The book titled "Natural Language Processing with Python" by Bird, S. et al. (2) gives an in-depth idea about the natural language processing toolkit that is available for Python. The book provides tremendous guidance for the first few modules: finding the most frequent words, frequent collocations, WordNet synsets, etc. There are plenty of Internet resources for gaining knowledge about the various tools available for summarization.

For classification, the article (3) on Apache Mahout gives information on the underlying implementation as well as step-by-step command line instructions for carrying out Naive Bayes as well as Complemented Naive Bayes.

For clustering, a research paper (4) on Latent Dirichlet allocation (LDA), a generative probabilistic model for collections of discrete data such as text corpora, introduces the concept and provides an in-depth understanding of the theory. Another resource (5) on Apache Mahout describes K-means and steps for carrying out pre-clustering and converting the clusters into a readable format.

For the abstractive summary, the paper "Get To The Point: Summarization with Pointer-Generator Networks" (6) discusses the approach of pointer-generator for automatic summarization along with the link to their source code on GitHub (7).

4 Approach and Implementation

Our team opted for the machine learning approach along with some deep learning techniques that we plan on applying later in the project. We have not chosen the deep learning only approach. These can be split into separate sections, or in the case of the 10 step approach, can be discussed for each of the steps. These explain what was accomplished, including functionality, and include: approach, tools, methodology, conceptual background, and deliverables. The entire process was broadly divided into several tasks or units. This made it easy for us to learn step by step and thus produce the results.

4.1 Unit 1: A set of most frequent important words

4.1.1 Introduction

Our first task was to extract the summary using the most common frequent words from the given dataset about the Tunisian election. The first obstacle that we faced was gathering the sentences_t part of JSON files into one string variable. To be able to do so, our team scanned all of the JSON files by for loops and sort the portions by using *python.json* package. After storing all contexts into the string, the script chunks strings into words. With the given stop words set from NLTK, it will automatically filter articles such as a, an, the, or they. Stop words are removed to save space and time to process large datasets. Unfortunately, since the NLTK stop words do not contain punctuation, the script manually filters words containing any special characters. After the filtering, the script calls most_common_words and most_common_collocations from the NLTK package.

4.1.2 Results

Table 1 shows the top 10 most frequently used word in articles from the Tunisia dataset. Table 2 shows the top 10 most frequently used collocations in articles from the Tunisia dataset. On similar analysis with the Maryland shooting dataset, we get results as shown in Figures 1, 3 for newsroom articles and Figures 2, 4 for school articles.

4.1.3 Challenges Faced

During this phase, we had a problem with compatibility between our local settings and the Hadoop cluster settings. We have used pip -user, adding path into /.bashrc file, and loading modules to avoid authorization issues. Also, during unit two our given topic changed from Tunisian Elections into Maryland Shooting. Most of the code stayed the same except for file paths and methods to load file contexts. We have compared multiple corpora from NLTK packages such as Brown corpus, Reuters corpus, and Gutenberg Corpus. It was hard to decide which corpus to choose since results would vary depending on the purpose of classification.

Table 1: The top 10 most common words

Rank	Word	Frequency
1	elections	1670
2	tunisia	1570
3	elections	1559
4	political	1082
5	party	1007
6	tunisian	760
7	new	757
8	parties	731
9	electoral	642
10	world	619

Table 2: The top 10 most common collocations

Rank	Collocation	Frequency
1	(u'Carter', u'Center')	1670
2	(u'Constituent', u'Assembly')	1570
3	(u'Ben', u'Ali')	1559
4	(u'Tunisia', u'Live')	1082
5	(u'Tunisia', u'world')	1007
6	(u'aim', u'connect')	760
7	(u'connect', u'Tunisia')	757
8	(u'independent', u'verified')	731
9	(u'offer', u'independent')	642
10	(u'world', u'offer')	619

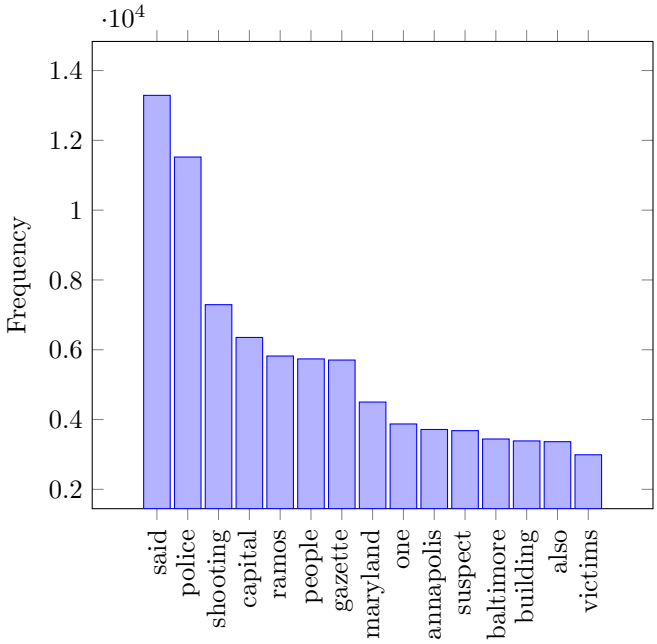

Figure 1: Most frequent words in Newsroom shooting

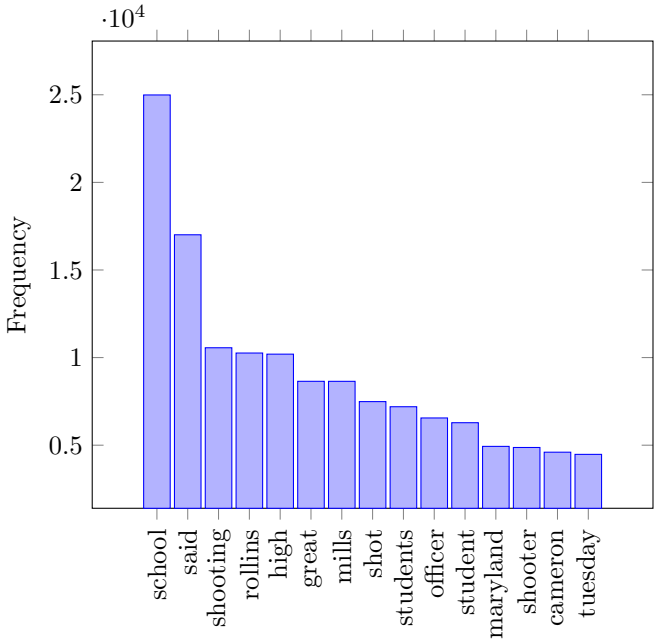

Figure 2: Most frequent words in School shooting

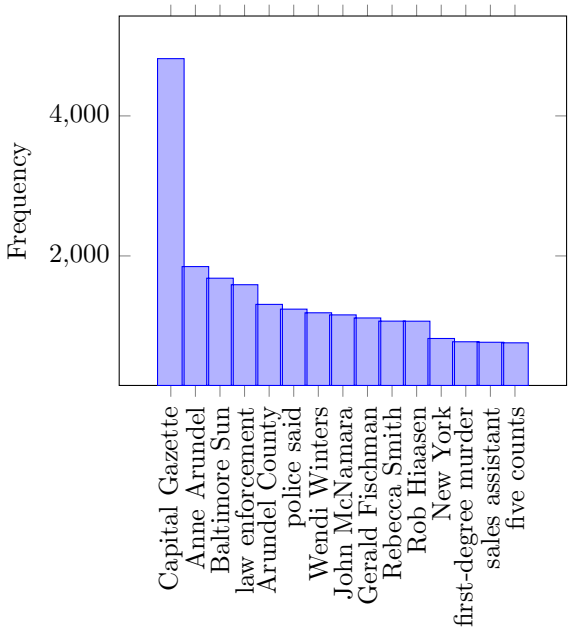

Figure 3: Most frequent bigrams in Newsroom shooting

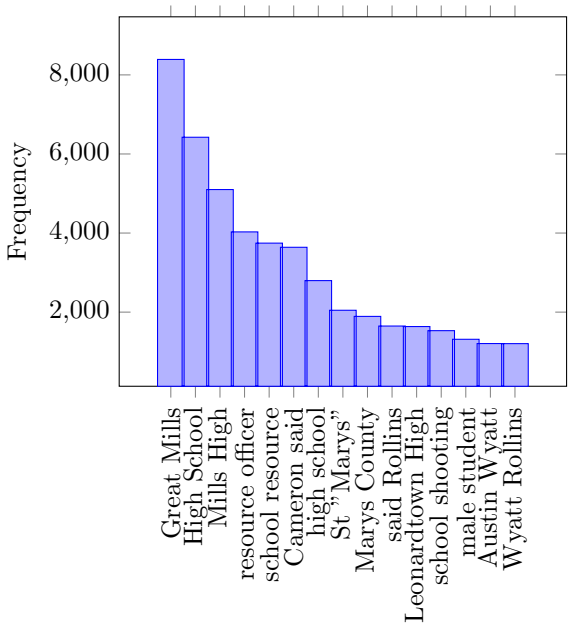

Figure 4: Most frequent bigrams in School shooting

4.2 Unit 2: A set of WordNet synsets that cover the words

4.2.1 Introduction

During Unit 2, our team computed the Automated Readability Index (ARI), generated the set of WordNet synsets and hypernyms that covers the words, and ranked synsets based on frequency. Automated Readability Index is used to determine the level of readers based on age levels. After the calculation, the ARI score of the target dataset is 13.083. According to table 3, this score means that the primary reader for those articles are college students or those who age between 18 -24.

Table 3: Automated Readability Index

Score	Age	Grade Level
1	5 - 6	Kindergarten
2	6 - 7	First/Second
3	7 - 9	Third
4	9 - 10	Fourth
5	10 - 11	Fifth
6	11 - 12	Sixth
7	12 - 13	Seventh
8	13 -14	Eight
9	14 - 15	Ninth
10	15 -16	Tenth
11	16 - 17	Eleventh
12	17 - 18	Twelfth
13	18 -24	College student
14	24+	Professor

According to the Table 3, the readability measures are targeted at college students. Our team implemented nltk.corpus.reader.wordnet to find the similarity sets from the most frequently used words from unit 1. The function synsets groups similar words together and calculates the frequency.

4.2.2 Results

List of hypernyms and hyponyms from Newsroom shooting:
set(['say', 'tell']),
set(['law', 'police']),
set(['shoot', 'buck', 'shot', 'sprout', 'take', 'pip', 'fool_away', 'shooting']),
set(['majuscule', 'cap', 'working_capital', 'capital']),
set(['the_great_unwashed', 'people']),
set(['gazette']),
set(['unity', 'one']),

set(['suspect', 'surmise']),
set(['dupe', 'victim']),
set(['building', 'make', 'edifice', 'build', 'ramp_up', 'progress', 'establish']),
set(["ship's_officer", 'officer'])

Table 4: Synsets generated from Newsroom Shooting

Synsets	Frequency
[u'said', u'read', u'say', u'articulate', u'order', u'aforesaid', u'state', u'suppose', u'aforementioned', u'allege', u'tell']	20086
[u'patrol', u'law', u'police']	14233
[u'shoot', u'buck', u'photograph', u'shooting', u'hit', u'fool', u'tear', u'sprout', u'flash', u'dissipate', u'dart', u'dash', u'charge', u'take', u'snap', u'shot', u'blast', u'film', u'inject'])) (7442, set([u'great', u'cap', u'washington', u'capital']	13659
[u'mass', u'multitude', u'citizenry', u'people']	6668
[u'gazette']	5707
[u'md', u'maryland']	5183
[u'ace', u'unity', u'unmatched', u'one', u'1', u'single']	5013
[u'funny', u'shady', u'suspicious', u'mistrust', u'distrust', u'suspect', u'defendant', u'surmise']	4040
[u'building', u'make', u'edifice', u'construct', u'construction', u'build', u'progress', u'establish']	4706
[u'officer', u'policeman']	3436

List of hypernyms and hyponyms from School shooting:
set(['school', 'civilise', 'schooling', 'school_day', 'shoal', 'schoolhouse']),
set(['say', 'tell']),
set(['educatee', 'student']),
set(['shoot', 'buck', 'shot', 'sprout', 'take', 'pip', 'fool_away', 'shooting']),
set(['high', 'high_school', 'heights']),
set(['mill', 'pulverisation', 'milling_machinery', 'manufactory', 'mill_around']),
set(['great']),
set(['shoot', 'buck', 'gibe', 'shot', 'sprout', 'dead_reckoning', 'shooter', 'stab', 'take', 'crack', 'pip', 'fool_away', 'pellet']),
set(["ship's_officer", 'officer']),
set(['shooter', 'crapshooter']),
set(['tues']),

set(['imagination', 'resource'])

Table 5: Synsets generated from School shooting

Synsets	Frequency
[u'school', u'train', u'educate', u'shoal']	26260
[u'shoot', u'buck', u'photograph', u'shooting', u'hit', u'tear', u'flash', u'dash', u'charge', u'take', u'fool', u'shot', u'blast', u'film']	21004
[u'high', u'high-pitched']	10211
[u'mill', u'factory']	8676
[u'heavy', u'great', u'bully', u'outstanding', u'big', u'swell', u'large', u'cracking', u'capital']	9283
[u'officer', u'policeman']	7536
[u'md', u'maryland']	5937
[u'gunman', u'shooter', u'shot', u'gun']	19027

4.3 POS Tagging

4.3.1 Introduction

A part of speech is a category of words assigned by its syntactic function. Part of speech tags will enable to identify the role of the target word in sentences. There are 35 tags in NLTK POS tagging as described in Table 6.

POS tagging distinguishes between tenses for verbs and plural and singular nouns. Therefore, the tags give a reader a better understanding of the usage of words. Our team applied POS tagging into the most common words of the small dataset of Maryland shooting.

We followed the steps below to generate the POS tagged words:

1. Tokenizing & Lemmatizing - Breaking the collection into its individual words and lemmatizing using NLTK's WordNet Lemmatizer.

2. Generating POS tags using NLTK's pos_tag method for the most frequent words found in Section 4.1.

4.3.2 Results

Tables 7 and 9 show the results of POS tagging the most frequent words found in the Newsroom shooting and School shooting collections. Tables 8 and 10 show the most frequent Nouns and Verbs found in both collections.

4.3.3 Challenges

Note that some of the common words are not correct. For example, 'trump' in the context meant the last name of President Donald Trump, but it tagged as the 'VB' which means present tense verb. Because of the limitations of wrongfully tagged words, it may cause further complications while using the part of speech tagging. There is no clear solution for the missing tagging issues; for now, the readers are supposed to read manually and understand the context to distinguish the misguidedly tagged words.

Table 6: POS tagging labels

Tag Name	Description
CC	coordinating conjunction
CD	cardinal digit
DT	determiner
EX	existential there
FW	foreign word
IN	preposition/ subordinating conjunction
JJ	adjective
JJR	adjective, comparative
JJS	adjective, superlative
LS	list marker 1
MD	modal could
NN	noun, singular
NNS	noun plural
NNP	proper noun, singular
NNPS	proper noun, plural
PDT	predeterminer
POS	possessive ending
PRP	personal pronoun
PRP	possessive pronoun
RB	adverb
RBR	adverb, comparative
RBS	adverb, superlative
RP	particle
TO	to
UH	interjection
VB	verb, base form
VBD	verb, past tense
VBG	verb, gerund/present participle
VBN	verb, past participle
VBP	verb, sing. present, non-3d
VBZ	verb, 3rd person sing. present
WDT	wh-determiner which
WP	wh-pronoun who, what
WP	possessive wh-pronoun whose
WRB	wh-abverb where, when

Table 7: POS tagging for most frequent words in Newsroom shooting

Word	POS tagging
'said'	'VBD'
'police'	'NNS'
'shooting'	'VBG'
'capital'	'NN'
'ramos'	'NN'
'people'	'NNS'
'gazette'	'VBP'
maryland'	'VBP'
'one'	'CD'
'suspect'	'NN'
'annapolis'	'NN'
u'victim'	'NN'
'building'	'NN'
'baltimore'	'NN'
u'officer'	'NN'
u'trump'	'VB'

Table 8: Most frequent Nouns and Verbs in Newsroom shooting

POS	Words
Verbs	'said', 'shooting', 'trump'
Nouns	'police', 'capital', 'ramos', 'people', 'suspect', 'annapolis', u'victim', 'building', 'baltimore', u'officer'

Table 9: POS tagging for most frequent words for School shooting

Word	POS tagging
'school'	'NN'
'said'	'VBD'
'student'	'NN'
'shooting'	'VBG'
'rollins'	'NNS'
'high'	'JJ'
u'mill'	'NN'
great'	'JJ'
u'shot'	'JJ'
'officer'	'NN'
'maryland'	'NN'
'shooter'	'NN'
'cameron'	'NN'
'tuesday'	'NN'
'resource'	'NN'
u'trump'	'VB'

Table 10: Most frequent Nouns and Verbs in School shooting

POS	Words
Verbs	'said', 'shooting', 'trump'
Nouns	'school', 'student', 'rollins', u'mill', 'officer', 'maryland', 'shooter', 'cameron', 'tuesday', 'resource'

4.4 Document classification

4.4.1 Approach

As seen in the preliminary K-means clustering of documents, we found that our collection contained documents of three types:

- Newsroom shooting
- School shooting
- Irrelevant

The documents in our collection needed to be classified into the above classes to get the most relevant information out of the corpus.

Naive Bayes has long been a standard in text classification. Complement Naive Bayes (CBayes), being an extension of Naive Bayes, performs particularly well on datasets with skewed classes and has been shown to be competitive with algorithms of higher complexity such as Support Vector Machines. The CBayes model is an implementation of Transformed Weight-normalized Complement Naive Bayes as introduced by Rennie et al. (8).

Since binary classifiers perform better, we decided to predict two classes at a time. Logistic regression is a popular binary classification model used for prediction of the probability of occurrence of an event. It makes use of several predictor variables that may be either numerical or categorical, that would help in predicting whether a document belongs to one of the two classes.

Manual labeling: We manually labeled 500 articles from our collection and evenly distributed the classes. In the end, we used 300 labeled articles with 100 belonging to each one of the three classes.

We tried the following classification approaches:

1. Complement Naive Bayes Model
   (a) With 3 classes :
   - newsroom_shooting, school_shooting, irrelevant

   (b) With 2 classes for each event:
   - newsroom_shooting, irrelevant (containing unnecessary and school related articles)
   - school_shooting, irrelevant (containing unnecessary and newsroom related articles)

2. Binary Logistic Regression Model
   (a) With 2 classes :
   - shooting (containing both shooting events), irrelevant
   - school_shooting, newsroom_shooting

4.4.2 Results

Table 11 below, displays the classification accuracy that was obtained using the CBayes as well as the logistic regression model.

Table 11: Accuracy obtained with various classification approaches

Classification model	Classes	Accuracy
CBayes Model	newsroom_shooting, school_shooting, irrelevant	76.84%
	newsroom_shooting, irrelevant	83.04%
	school_shooting, irrelevant	90.22%
Logistic Regression Model	shooting, irrelevant	81.73%
	school_shooting, newsroom_shooting	97.67%

4.5 Named Entities

4.5.1 Approach

Named Entity Recognition is one of the most important steps in NLP. This is because it helps in identifying and classifying named entities into persons, organizations, locations, etc. Named Entity extraction can help in answering some of the questions related to the articles, for example: Who was the shooter? Where did the shooting take place? We used NLTK's ne_chunk() method. Since we did not get the best results, we decided to use SpaCy's Named Entity Recognition.

We followed the steps below to identify named entities:

1. Tokenizing - Breaking the collection into its individual words.
2. NE Tagging using NLTK's NER - Classify the most frequent words found in Section 4.1 using NLTK's ne_chunk method.
3. NE Tagging using SpaCy's NER - Classify the most frequent words and bigrams found in Section 4.1 using SpaCy's Named Entity Recognition.

4.5.2 Results

Table 12 shows the NER results using NLTK for the school shooting event where as Table 13 shows the result for the newsroom shooting event.

Table 12: NER using NLTK for School shooting

Word	Named Entity Tag
school	NN
said	VBD
student	NN
shooting	VBG
Rollins, Mills, Great	PERSON - NNP
shot	JJ
officer	NN
School, High, Maryland	PERSON - NNP
shooter	NN
Cameron	PERSON - NNP
Tuesday	NNP
resource	NN
gun	NN
parent	NN
fired	VBD
Gaskill	PERSON- NNP

Table 13: NER using NLTK for Newsroom shooting

Word	Named Entity Tag
said	VBD
shooting	JJ
police	NNS
Capital Ramos	ORGANIZATION- NNP
Gazette	NNP
people	NNS
Police	NNP
Maryland	NNP
suspect	NN
victim	NN
Annapolis	PERSON- NNP
building	NN
Baltimore	NNP
one	CD
shot	NN
year	NN
also	RB
scene	VBD
say	VBP

From Tables 12 and 13 it is clear that Named Entity Recognition using NLTK works well but also misclassifies certain words so we decided to try SpaCy for the same purpose.

Table 14 and 15 displays the result of Named Entity Recognition using SpaCy.

Table 14: NER using SpaCy for Newsroom shooting

Word	Named Entity Tag
first	ORDINAL
Baltimore Sun	PERSON
Annapolis	GPEM
Maryland	GPE
Capital Gazette	ORG
Arundel County	GPE
Rob Hiaasen	PERSON
New York	GPE
Jarrod Ramos	PERSON
Capital	ORG
Wendi Winters	PERSON
Gazette Annapolis	PERSON
Anne Arundel	PERSON
Gerald Fischman	PERSON
Rebecca Smith	PERSON
John McNamara	PERSON
five	CARDINAL
June 28	DATE

Table 15: NER using SpaCy for School shooting

Word	Named Entity Tag
Tuesday	DATE
morning	TIME
Mills	PERSON
Leonardtown High	PERSON
Maryland	GPE
Blaine Gaskill	PERSON
Rollins	ORG
Great Mills	ORG
20 2018	DATE
Tim Cameron	PERSON
Austin Wyatt	PERSON
14-year-old	DATE
Jaelynn Willey	PERSON
March 20	DATE
Mary	PERSON
Rollins	PERSON
St Marys	PERSON
Wyatt Rollins	PERSON
Cameron	ORG
Mills High	PERSON

It is clear from the results of Tables 14 and 15 that SpaCy does a much better job when it comes to tagging the words.

4.6 LDA

4.6.1 Approach

The unsupervised learning algorithm Latent Dirichlet allocation (LDA) (4) is used on our data collection for topic extraction. In LDA, the composites are documents and the parts are words. Each document is viewed as a mixture of various topics and is assumed to be characterized by a particular set of topics. Using LDA is a way of soft clustering of composites and parts by viewing the number of topics as the number of clusters and the probabilities as the proportion of cluster membership.

We first split each document into sentences using NLTK tokenizer, then ran the LDA topic model on our big dataset.

Three topics are found in our dataset. The first one is for the newsroom shooting which contains the location, Capital Gazette, and the shooter's name, Ramos. The second topic is also for the newsroom shooting but with more details on the victims. And the third topic is for school shooting which involves the school name, Great Mill high school, and details on students and officers.

4.6.2 Results

Table 16 describes the LDA topics obtained and their corresponding probabilities.

Table 16: LDA Topics with their corresponding probabilities

Topic	Probabilities
Topic 1	0.011*"capital" + 0.010*"shooting" + 0.008*"gazette" + 0.006*"news" + 0.006*"newspaper" + 0.006*"said" + 0.005*"one" + 0.005*"ramos" + 0.004*
Topic 2	0.011*"said" + 0.008*"capital" + 0.007*"shooting" + 0.006*"gazette" + 0.006*"police" + 0.004*"news" + 0.004*"people" + 0.004*"Maryland" + 0.004*"victims"+ 0.004*"newspaper"
Topic 3	0.019*"school" + 0.011*"high" + 0.010*"police" + 0.008*"great" + 0.008*"said" + 0.008*"mills" + 0.006*"students" + 0.006*"said." + 0.006*"shooting" + 0.005*"officer"

4.6.3 Challenges

It was not hard to tokenize articles into sentences by applying code provided on Canvas, but it was hard to train the LDA model well enough to extract topics because of the noise in our dataset. The keyword 'shooting' is contained in all 3 topics, which makes them so similar to each other. There are also some stopwords like 'said' in topics which do not contribute to the understanding of each topic.

4.7 K-means

4.7.1 Approach

K-means clustering is a simple and fast hard clustering method for machine learning. K-means firstly randomly picks the initial centroids for each cluster(k). Then it will determine the point that is closest to the clusters' centroids, and use the points in the cluster to compute the new centroid. The result is then produced by repeating these steps for a specified number of iterations.

We use Mahout's K-means on our dataset. We followed the below steps to carry out K-means clustering:

1. Chunked the articles into sentences
2. Converted the sentences into a ⟨ Text, Text ⟩ SequenceFile.
3. Converted and preprocessed the dataset into a ⟨ Text, VectorWritable ⟩ SequenceFile containing term frequencies for each document.
4. Used canopy clustering to compute the initial clusters for K-Means.
5. Generated the K-means clusters.

4.7.2 Results

We set k=10 for our small data set to see what the result would look like. We get Table 17 for newsroom shooting and Table 18 for school shooting.

Table 17: Newsroom Shooting K-means Results

EVENT	RESULT
Newsroom Shooting	centroid: ['walking:6.867', ' annapolis:6.042', ' gazette:3.786', ' vigil:6.673', ' hundreds:7.021', ' were:3.469', ' md:5.265', ' attended:7.634', ' five:4.593', ' newsroom:4.788', ' mourn:9.101', ' employees:5.604', ' shooting:3.274', ' death:6.025', ' city:5.882', ' who:3.704', ' during:5.202', ' capital:3.718', ' thursday:4.662', ' people:3.983', ' killed:4.440']
Newsroom Shooting	centroid: ['bill:7.108', ' ramos:5.781', ' annapolis:4.272', ' gazette:3.786', ' june:5.137', ' inside:5.143', ' court:4.921', ' suspect:4.735', " here's:8.002", ' suspected:6.040', " suspect's:7.203", ' district:6.393', ' 29:5.792', ' hearing:8.337', ' arundel:5.143', ' anne:5.093', ' sketch:12.463', ' maryland:3.735', ' captured:8.120', ' scene:4.801', ' shooting:3.274', ' shooter:4.385', ' jarrod:7.529', ' 2018:4.601', ' during:5.202', ' bail:9.279', ' capital:3.718', ' et:6.733', ' courtroom:12.870', ' from:3.416']

Table 18: School Shooting K-means Results

EVENT	RESULT
School Shooting	centroid: ['attacked:7.381', ' state:5.461', ' gunman:4.481', ' shot:5.620', ' unsupported:8.423', ' school:3.202', ' new:4.725', ' panel:8.605', ' your:4.661', ' police:3.472', ' external:7.649', ' links:7.170', ' window:7.912', ' media:6.831', ' injured:4.978', ' gunfight:8.135', ' maryland:3.724', ' our:4.541', ' dead:5.252', ' officer:4.280', " caption'greatest:9.116", ' device:7.219', ' two:4.272', ' us:5.252', ' playback:8.423', ' teenage:7.270', ' after:3.666', ' high:3.953', ' open:6.996', ' children:5.444', ' who:3.707', ' close:6.631', ' students:4.211', ' nightmare:8.828', ' share:5.784']
School Shooting	centroid: ['campus:6.386', ' douglas:5.995', ' have:4.112', ' rocks:8.135', ' garden:8.605', ' where:4.803', ' been:4.045', ' memorial:6.207', ' off:5.570', ' 17:4.872', ' half:6.688', ' kept:7.730', ' lost:7.036', ' marjory:6.386', ' stoneman:6.137', ' those:5.609']

4.7.3 Conclusion

From the example results above, we find out that the results of both shooting events are not satisfying. Both results have some words that are related to the shooting event, such as place names (annapolis, newsroom), people names (douglas, ramos), and some other words related to shooting. But the results also have many words that are not helpful, such as 'where', 'been', 'who' and 'after'.

4.7.4 Challenges Faced

We found the number of clusters that was initially defined was hard to determine. We tried to specify k values like 10 and 20. The results are different but it's hard to tell which one is better. Then we did not specify the number of clusters but instead used Mahout's canopy method to determine the number of clusters for us. But it's still hard to tell whether this way is better then specifying the k value. All the results contain useful centroid result and unhelpful results. We think it's because we have 2 shooting events and our classified sentences still have much noise. Some stop words also appear in the result and are not helpful for us.

4.8 Regular Expression Extraction

4.8.1 Approach

We used a semantic slot matching approach to obtain an extractive summary which is explained in detail in section 4.10. Semantic Slot Matching (9) is a technique where a template is filled using details from the dataset. We used regular expression in Python scripts to obtain the answers to the questions and then we used those results to fill up the details of the template to obtain the summary. An example of a Python regular expression follows.

4.8.2 Results

An example of regular expression used to obtain the summary.

```
re.search( r'([^.]*?( identified)[^.]*(?=shooter|)
[^.]*\.)', line, re.I)
```

The regex above attempted to find the name of the shooter. The results were:

For newsroom shooting:
"The suspected gunman has been identified as Jarred Warren Ramos, 38, who is a resident of Laurel, Maryland"
"The suspect was identified as Jarrod Ramos, 38, three senior law enforcement officials briefed on the matter told NBC News."

For school shooting:
"Identified Austin Wyatt Rollins as the shooter"

The regex performed well in extracting the shooter's name. Similar regular expressions were applied to obtain other details about the shooting instances. Due to the size of the collection, the regex search returned multiple results. We selected the result with the highest frequency as the value for a slot. Specifically for names, locations, etc., we used SpaCy's named entity recognition on the regex results as these details were difficult to identify using only regular expressions.

4.8.3 Challenges Faced

Our primary challenge in all of our approaches was to deal with the two instances of Maryland Shooting. In spite of prior document classification, the regex extraction resulted in having some data from the other shooting event as well. This problem was dealt with by filtering out the regex expression by using keywords specific to the instances.

4.9 Extractive summary using slot semantic matching

4.9.1 Approach

To generalize filling a template for any kind of shooting event, it was necessary to develop a context-free grammar. A context-free grammar is a set of production rules that describe all possible strings in a given formal language. [wiki ref https://en.wikipedia.org/wiki/Context-free$_grammar$].

We first constructed a structure for our summary using the shooting-specific template and then developed a CFG as depicted in table 19

Table 19: Context Free Grammar for Summary

Non-terminals	Terminals
Summary	Intro Details AdditionalDetails Aftermath
Intro	On Date, ShooterDetails, opened fire at Location at Time.
Details	Killed and Injured before Climax. AgeRange Respondents
AdditionalDetails	WeaponInformation Motive
Aftermath	Resolution PolicyChanges
Date	Month Number,Year
ShooterDetails	ShooterName, Number
Killed	The shooter killed Number
Injured	Number—several were left wounded
Climax	the shooter was caught | the shooter shot himself
AgeRange	The victims were aged between Number and Number
Respondents	Number respondents arrived at—in Time
WeaponInformation	The shooter fired Number rounds using Words
Motive	The motive behind this attack was Words
Resolution	ShooterName faced Words
PolicyChanges	Words
ShooterName	Word Word
Words	Word Words
Word	[a-zA-Z]+
Number	[0-9]+

4.9.2 Results

Based on the above grammar we were able to generate the following two extractive summaries for the two shooting events:

Newsroom Shooting:
On June 28, 2017, Jarrod Ramos, 38 opened fire at Capital Gazette, Maryland at 2:34 p.m. The shooter killed 5 and several were left wounded before getting caught. The victims were aged between 34 and 65. County Police arrived in 1-minute.

The shooter fired some rounds using a 12-gauge pump-action shotgun. The motive behind this attack as he had a previous dispute with the newspaper. Ramos faced a grand jury indictment within the next 30 days. In the wake of that attack and the general rise in threats, many newsrooms have stepped up emergency drills and taken other measures to increase security.

School Shooting:
On March 20, 2018, Austin Wyatt Rollins, 17 opened fire at Great Mills High School, in St. Mary 's County, MD at 7.55 am in the morning. The shooter killed one and was left wounded. The victims were aged between 14 and 17.

The shooter fired one round using a semi-automatic Glock. The shooter had ended his relationship with the girl he shot and that is believed to be the motive. Moments after the gunfire, a school official was alerted, and he fired around in order to engage the shooter. Austin was killed by the officer's bullet. The students asked for further school safety measures and called for an end to gun violence.

4.10 Abstractive summary using Pointer-generator

4.10.1 Approach

We used deep learning mechanisms to obtain an abstractive summary from our dataset. An abstractive summary, as opposed to the extractive summary, may generate novel words and phrases.

We used the "Get To The Point: Summarization with Pointer-Generator Networks" (6). The traditional neural sequence-to-sequence models have a few shortcomings; they often reproduce incorrect factual details and tend to repeat themselves. The Pointer Generator method has the ability to reproduce words from the original text via *pointing*; this is what helps in the reproduction of text. The generator helps to produce novel words. We followed the steps mentioned in the GitHub repository `https://github.com/chmille3/pointer-generator`. Basically, the idea is to use a pre-trained model, that has been trained on the CNN/Daily Mail dataset.

The steps that we followed to get a human-readable summary from the articles are as follows:

1. We chunked the big collections (both newsroom and school events) into individual sentences and carried out K-Means clustering with cosine similarity distance measure.
2. We downloaded the Stanford CoreNLP that was required to tokenize the data.
3. We processed the clustered data to .bin and vocab files.
4. We downloaded the pre-processed CNN/DailyMail dataset for the vocab file and the pre-trained model.
5. We ran the pointer-generator on the processed clustered sentences to summarize each of the clusters.

4.10.2 Results

On running the pointer-generator, 2-3 sentences were generated from each of the clusters which summarized those clusters. After some manual post-processing of the collection of sentences, which involved punctuation, capitalization, and ordering of sentences to maintain coherency, we obtained the following two summaries for the shooting events:

Newsroom shooting:
5 dead in 'targeted attack' at Capital Gazette newspaper on June 28, 2018, in Annapolis, Maryland, with many others wounded. The suspect was identified in reports as Jarrod Ramos. There were about 170 people in the paper's Annapolis office when a gunman shot through a glass door, opening fire on multiple employees. He is accused of entering the capital gazette office, firing through a

glass door with a 12-gauge shotgun, hunting for victims. Phil Davis, a capital crime reporter who was in the building at the time of the shooting, said multiple people were shot as he and others hid under their desks. Rob Hiaasen, Wendi Winters, Rebecca Smith, Gerald Fischman, and John McNamara were shot and killed Thursday afternoon. Two other employees, Janet Cooley, and Rachel Pacella were treated for minor injuries. John McNamara, 56, was a reporter for the Capital Gazette who focused on Bowie, Crofton-West County. Ramos, who lives in Laurel, MD., has a criminal conviction in his past, including criminal harassment. Ramos filed a defamation suit against the paper in 2012. News4's Jackie Bensen reports on the history of threats the shooting suspect made to a woman and then the newspaper in past years. He is known to have had a conflict with the Capital Gazette newspaper. It was revealed that Ramos sued the Capital Gazette in 2012 for defamation. The new attempted murder charge alleges Ramos tried to kill photographer Paul Gillespie. Authorities have not released any information as to a motive for the attack. But Krampf said that they were investigating threats sent to the Capital Gazette through social media as recently as today. Ramos, 38, was swiftly arrested as he tried to hide under a desk Thursday afternoon. The grand jury indicted Jarrod Ramos, 38, of Laurel, on five counts of first-degree murder in connection with the deaths of five employees of the capital gazette in Annapolis. It was one of the deadliest attacks on journalists in U.S. History.

School shooting:
Austin Rollins, a student at Great Mills High School in St. Mary's County, MD, started firing in a hallway at the start of the school day. The shooting broke out just before classes were scheduled to start at Great Mills High, a 1,500-student school 65 miles south of Washington, D.C. All of this occurred before calls were placed to 911 starting at 7:58 a.m., just before classes started. Tyriq Wheeler, 17, was headed to English class when he heard a bang. Isaiah Quarles, a 10th-grader, was walking to his first-period class on Tuesday. A student named Jonathan Freese called into CNN and said the shooting began early in the morning and seven people could possibly be hurt. The entire incident played out in less than a minute at 7:55 a.m. in a hallway at Great Mills, a school 90 miles south of Baltimore. A student witness of the school shooting in Southern Maryland says a police officer tried to order a student with a gun to his head to disarm before two shots were fired. The lockdown was announced once he had made it to the classroom, where blinds were lowered and the door locked. The 17-year-old who opened fire on classmates at Great Mills High School in Southern Maryland last week, injuring one and killing another, died from shooting himself in the head. The attack ended after Gaskill ran inside and confronted Rollins, with each firing a single shot almost simultaneously. Rollins was confirmed dead nearly three hours later at a hospital. The sheriff initially told reporters Rollins shot both victims with a handgun, but it's not yet clear whether Rollins was killed by the officer's bullet or from a self-inflicted wound. Gaskill, who was not injured, followed protocol, Cameron added. Authorities initially said Rollins shot the male victim after shooting the female victim. He

shot Jaelynn Willey, 16, in the head, and the bullet also struck a 14-year-old boy, Desmond Barnes, in the leg. The wounded girl, 16-year-old Jaelynn Willey, remains in critical condition at The University of Maryland Prince George's hospital center. Rollins had a prior relationship with the girl, Sheriff Tim Cameron said. Mason said Rollins was a quiet kid who was never disrespectful. It came a month after 17 people were killed at a Florida high school. The latest in a long string of deadly shootings at U.S. schools and colleges occurred little more than a month after 14 students and three faculty members were fatally shot on Feb. 14. Great Mills students joined others across the country to protest gun violence at schools during the national school walkout on Wednesday. The survivors are organizing a national demonstration for gun control laws on March 24 called March for Our Lives.

4.10.3 Challenges Faced

We faced a few challenges while following the steps of the Github repository. First and foremost, we had some problem getting the Stanford CoreNLP tokenizer to work. It was resolved by exporting CLASSPATH to the right folder.

Secondly, we ran into some trouble while running Step 3 from the steps mentioned above. It was primarily because we were not using the vocab files of the pre-trained model. Using the correct vocab file helped us obtain the summary of the articles as discussed in the result section.

5 Evaluation

5.1 Golden Summary for Maryland Shooting

Golden Standard summary as prepared by Team 14.

Newsroom Shooting:

On Tuesday, June 28, 2018, at 2:33 pm, 38-year-old Jarrod Warren Ramos opened fire on the glass front door of the Capital Gazette newsroom in Annapolis, MD. Of the 11 employees there that day, 5 were killed: John McNamara, Gerald Fischman, Rob Hiaasen, Wendi Winters, and Rebecca Smith. Two more were injured. Ramos used a 12 gauge pump-action shotgun and smoke grenades and barricaded the back door to the newsroom to prevent people from escaping. One Capital Gazette employee, Wendi Winters, grabbed a trash can and a recycling bin and charged the gunman, but was killed. Law enforcement responded within 60 seconds and arrested the gunman, who was found hiding under a desk. Approximately 170 people were evacuated from the building.

The Capital Gazette is a small local newspaper owned by the Baltimore Sun. In 2011, the Capital Gazette reported on a case in which Ramos was convicted of harassing a former high school classmate. Ramos sued Eric Hartley and Thomas Marquardt, employees of the Capital Gazette, for defamation in 2012, and the lawsuit was dismissed. Ramos began to make threats against the Capital Gazette. Editor Thomas Marquardt called the police to report the threats in 2013, but nothing came of it.

Ramos has been charged with 23 total charges including five counts of first-degree murder and was held without bail. Ramos was assigned public defender William Davis. Davis claims that the facial recognition methods used to identify Ramos when his fingerprints were slow to process were impermissible. Ramos has pled not guilty to all charges and Davis has indicated that they are considering a plea of not criminally responsible by reason of insanity.

Several individuals associated with the Capital Gazette received letters postmarked the day of the shooting with Ramosís personal information and indicating a ìobjective of killing every person presentî, apparently signed by Ramos. The Capital Gazette also reported receiving politically-motivated threats after the initial coverage of the incident.

The Capital Gazette continued publishing the paper on schedule despite their loss. Several crowdfunding campaigns were set up for families, victims, and survivors of the attack, as well as a journalism scholarship memorial fund.

Shootings like these have sparked a call for tighter gun control regulations from some. A Maryland law was passed in April which gives family members and law enforcement the ability to temporarily restrict firearm access for individuals

believed to be a risk, but the law does not go into effect until October. Many questions whether the law could have been used against Ramos to prevent the attack.

After saying his ìthoughts and prayersî were with the victims, President Trump initially refused requests to fly the American flag half-mast but reversed his decision the Tuesday after the shooting.

Great Mills High School shooting

On Tuesday, March 20, 2018, at 7:57 a.m, 16-year-old Jaelynn Willey was shot in the halls of Great Mills High School in Great Mills, Maryland. The bullet that hit Willey also struck 14-year-old Desmond Barnes in the leg. Willey was rushed to the hospital and declared brain dead, then taken off of life support two days later after being declared brain dead. The shooter, Austin Wyatt Rollins, was a 17-year-old Great Mills student. Rollins used a Glock semi-automatic handgun to carry out the attack. The gun was legally owned by Rollinsí father, but it should not have been in Rollinís possession since Maryland law states that one needs to be over 21 to carry a gun. Following the shooting, Rollins wandered around the school. The school resource officer Deputy First Class Blaine Gaskill confronted Rollins and ordered him to drop his gun. 31 seconds after the confrontation, Rollins, and Gaskill fired their weapons simultaneously. Rollins fatally shot himself in the head and Gaskill shot him in the hand.

The school was put into lockdown for several hours following the shooting. Law enforcement arrived at the scene at approximately 8:00 a.m., but by then the situation had been contained. 1440 students were evacuated to a reunification center in nearby school Leonardtown High School. Classes were canceled for the rest of the week.

Investigators say that Rollins had a previous relationship with Jaelynn Willey that had recently ended. In the month before the shooting, the school had investigated a potential school shooting threat and determined that the threat was unsubstantiated. This was also less than a week after the students had participated in a nation-wide walkout to protest gun violence and the lack of gun regulations as well as show support for the victims of the Marjory Stoneman Douglas High School shooting. It was the 17th American school shooting in 2018. Following the shooting, school nurses have been trained in how to react to a school shooter, how to triage wounded children, and how to apply a tourniquet.

5.2 Rouge Scores

We compared our extractive and abstractive summaries with the golden standard summary and obtained the rouge scores for them. The rouge scores were calculated using a script provided by our Graduate Teaching Assistant.

The Rouge Paragraph Scores for Extractive and Abstractive summary as compared to the golden standard summary above are depicted in Tables 20 and 21 respectively.

Table 20: Rouge Scores for Extractive Summary

EVENT	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
Newsroom Shooting	0.33333	0.08571	0.25	0.13
School Shooting	0.26471	0.15152	0.23529	0.09574

From the above table it is clear that the Rouge-1 score for both the shooting instances were higher than other rouge paragraph scores. This suggests that we were able to capture most of the similar words as present in the Golden Summary. However, for the bigrams, the generated summary did not capture them as it did the unigrams.

Table 21: Rouge Scores - Paragraph for Abstractive Summary

EVENT	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
Newsroom Shooting	0.36111	0.11429	0.19444	0.1
School Shooting	0.20588	0.12121	0.17647	0.07979

Similar to the extractive summary, the abstractive summary also captures unigrams better than it performs in capturing the bigrams.

Below mentioned are the Rouge sentence level scores and the entity coverage for our abstractive summary. Table 22 and 23 gives a glimpse of how the predicted sentence from the pointer generator technique differs from the golden summary sentence.

Newsroom Shooting:

Max ROUGE-1 score among sentences: 0.73333

Thus, the entity coverage for newsroom shooting was calculated to be 38.10%

Table 22: Predicted vs. Golden Summary Sentence

Predicted Sentence	Rob Hiaasen, Wendi Winters, Rebecca Smith, Gerald Fischman and John McNamara were shot and killed thursday afternoon.
Golden Summary Sentence	Of the 11 employees there that day, 5 were killed: John McNamara, Gerald Fischman, Rob Hiaasen, Wendi Winters, and Rebecca Smith.

School shooting:
Max ROUGE-1 score among sentences: 0.88889

Table 23: Predicted vs. Golden Summary Sentence

Predicted Sentence	He shot Jaelynn Willey, 16, in the head, and the bullet also struck a 14-year-old boy, Desmond Barnes, in the leg.
Golden Summary Sentence	The bullet that hit Willey also struck 14-year-old Desmond Barnes in the leg.

Thus, the entity coverage for school shooting was calculated to be 27.27%

From the entity coverage scores for both the instances, it can be inferred that we were able to produce a third of the information of the events.

6 User Manual

This manual is for those who interested in re-running our experiments. The scripts below have been written to run on our dataset but can be used on other datasets as well with a few modifications to the paths or inputs used. First, we describe the manual for using Solr and then give a brief overview of the files that can be used for summarization along with their use cases.

6.1 Solr Manual

The following manual is aimed to help those who are interested in using Solr to find the information from our collection. Solr has its own tutorial (10) which we referred to for the following tasks:

- **Keyword Matching**
  Use 'AND', 'OR' and '-' between same or different field to find exact keywords.
- **Wildcard Matching**
  Use '*' after or between keyword for wildcard matching.
- **Proximity Matching** Use ' ' with integer value to indicate the distance between two keywords.
- **Range Searches**
  Use '[value] To [value]' pattern to find the fields' values that are in the range.
- **Boosts**
  Specify terms that are more important by applying higher boost factors to that term.

6.2 Files and Use Cases

Table 24 describes the files used for various summarization techniques along with the modules required and use cases. The steps for executing the files have been mentioned in the Developer Manual along with their execution times. All of our codes are posted in our repository: `https://github.com/BigDataTeam-15/summarization-maryland-shooting`.

Table 24: Program File Description

Files	**Required modules**	**Use case**
most_frequent.py	NLTK tokenizer, stop words	To find the most frequent words and bigrams.
synset_generation.py	NLTK tokenizer, stop words, WordNet synsets	To find the most frequent synsets, hyposets and hypersets.
pos_tagging.py	NLTK tokenizer, stop words, NLTK pos_tag	To find the most frequent nouns and verbs.
c_bayes_classify.sh	Hadoop, Mahout	To classify documents using Complement Naive Bayes. This script also includes conversion of documents into vectors and evaluation of classifier.
log_classify.sh	Hadoop, Mahout	To classify documents using Logistic Regression model. This script also includes conversion of documents into vectors and evaluation of classifier.
ner_tagging.py	NLTK tokenizer, stop words, NLTK pos_tag, SpaCy, en_core_web_sm	To find the most frequent named enitites using two modules - NLTK and SpaCy.
lda.sh, LDA-code.py	Hadoop, Mahout	To find the most important topics from the collection.
kmeans.sh	Hadoop, Mahout	To find the clusters containing similar content from the collection.
regex_extraction.py	re, SpaCy, en_core_web_sm	To find values to match semantic slots.
pointer_generator.sh	Stanford CoreNLP, pointer-generator	To generate a readable summary using pointer-generator.

7 Developer Manual

The following manual aims to help those interested in reproducing or extending our work. All of our codes are posted in our repository: `https://github.com/BigDataTeam-15/summarization-maryland-shooting`

7.1 Module Description

Following is a list of modules that are required along with installation details:

- **NLTK**
  pip install nltk (add –user if permission denied)
- **SpaCy**
  pip install spacy
  pip install `https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-2.0.0/en_core_web_sm-2.0.0.tar.gz` –user
- **Apache Mahout** (requires java as well)
  wget `https://www-us.apache.org/dist/mahout/0.13.0/apache-mahout-distribution-0.13.0-src.tar.gz`
  wget `ftp://apache.cs.utah.edu/apache.org/maven/maven-3/3.6.0/binaries/apache-maven-3.6.0-bin.tar.gz`
  export MAVEN_HOME=/path/to/apache-maven-3.6.0/bin
  cd apache-mahout-distribution-0.13.0
  $MAVEN_HOME/mvn -DskipTests -X clean install
  export MAHOUT_HOME=/path/to/apache-mahout-distribution-0.13.0/bin

7.2 Program File Description

The project software was developed in units, as discussed in Section 4, with the idea that we would have deliverables at certain checkpoints throughout the semester. The overview is as follows:

- Pre-processing dataset: We converted the WARC and CDX files to JSON format and then to individual text files for each article.
  - **Requirements:** None
  - **Files:** preprocess.sh, ArchiveSpark_sentence_extraction.scala, clean_files.py, keyword_classify.py
  - **Steps to execute:**
    1. ./preprocess.sh
    2. python2.7 clean_articles.py
    3. python2.7 keyword_classify.py
  - **Execution time:** 21.0810987949 seconds

- A set of most frequent important words: We found the most frequent words used from our dataset using NLTK. We also used the NLTK stopwords collection to filter out some unwanted words from the dataset. We found the top 100 frequent words and the results have been discussed in section 4.1.2.
  - **Requirements:** NLTK
  - **Files:** most_frequent.py
  - **Steps to execute:**
    1. python2.7 most_frequent.py
  - **Execution time:** 9.89873313904 seconds
- A set of WordNet Synsets that cover the words: This module mainly focused on finding out the similar words, namely the Synsets and calculating their frequency.
  - **Requirements:** NLTK
  - **Files:** synset_generation.py
  - **Steps to execute:**
    1. python2.7 synset_generation.py
  - **Execution time:** 33.7583019733 seconds
- A set of words constrained by POS, e.g., nouns and/or verbs: In this module, we identified part of speech tags for the most frequent words. E.g., police gets a NN tag which means noun, singular.
  - **Requirements:** NLTK
  - **Files:** pos_tagging.py
  - **Steps to execute:**
    1. python2.7 pos_tagging.py
  - **Execution time:** 12.991890192 seconds
- A set of words/word stems that are discriminating features (that also are helpful in a classifier for the relevant web pages): This module was a crucial one, where we classified the shooting instances into newsroom and school shooting instances and the irrelevant data.
  - **Requirements:** Apache Mahout
  - **Files:** c_bayes_classify.sh, log_classify.sh
  - **Steps to execute:**
    1. ./c_bayes_classify.sh
    2. ./log_classify.sh
  - **Execution time:** 224.2318945479 seconds

- A set of frequent and important named entities: In this module, we identified the named entities among the frequent words and frequent bigrams found.
  - **Requirements:** NLTK, SpaCy, en_core_web_sm
  - **Files:** ner_tagging.py
  - **Steps to execute:**
    1. python2.7 ner_tagging.py
  - **Execution time:** 15.8995079994 seconds
- A set of important topics, e.g., identified using LDA: This module used Latent Dirichlet Allocation to extract various topics from our dataset.
  - **Requirements:** Apache Mahout
  - **Files:** lda.sh
  - **Steps to execute:**
    1. ./lda.sh
  - **Execution time:** 294.4849373934 seconds
- A set of important sentences, e.g., identified by clustering: This module used K-means clustering to identify the clusters and to group similar topics together.
  - **Requirements:** Apache Mahout
  - **Files:** kmeans.sh
  - **Steps to execute:**
    1. ./kmeans.sh
  - **Execution time:** 238.8493028494 seconds
- A set of values for each slot matching collection semantics: We used regular expressions to obtain the answers to questions related to the shooting instances.
  - **Requirements:** SpaCy
  - **Files:** regex_extraction.py
  - **Steps to execute:**
    1. python2.7 regex_extraction.py
  - **Execution time:** 179.483909202 seconds
- A readable summary explaining the slots and values: This module utilized the results of the previous step to obtain an extractive summary.
  - **Requirements:** None
  - **Files:** slot_matching.py

  - **Steps to execute:**
    1. python2.7 slot_matching.py
  - **Execution time:** 72.482048594 seconds

- A readable abstractive summary, e.g., from deep learning: An abstractive summary was generated using the pointer generator technique of deep learning.
  - **Requirements:** Apache Mahout, pointer-generator
  - **Files:** abstractive_summarization.sh
  - **Steps to execute:**
    1. ./abstractive_summarization.sh
  - **Execution time:** 403.4894793034 seconds

8 Discussion and Conclusion

The biggest problem of summarizing big collections is the size. It is fairly easy to summarize a short article or a bunch of articles. But with a big collection, there is more irrelevant information and the problem of memory requirements. Therefore, it is necessary to clean the collection and reduce the size as much as possible without missing out on the important topics. Though we were able to eliminate noise in the collection, we believe carrying out the following would improve our results further:

1. **Being able to identify content completely irrelevant to shooting events at Maryland:** Our corpus contained a significant amount of information about other news facts at that time or information about shooting events at other places like Florida, etc. The collection also included shooting attacks that took place in Maryland, other than the ones we were focusing on. Though we tried classification to remove irrelevant documents, we were unable to build a classifier to pick out relevant sentences from articles. Further work can be done to build and train classifiers by manually labeling relevant and irrelevant sentences and using the classifier to extract only the most important and relevant sentences.

2. **Being able to eliminate duplicate articles:** We found that many articles from the collection had the same content. In spite of being from different sources, the content was duplicated, which only added to the time taken to process, and to redundancy. One quick way to check for duplicate content can be to hash articles and compare their hash values. But this would work only if the content is an exact duplicate. Another way could be checking for similarity between articles. If two articles have a very high similarity score (almost close to 1), we can call the articles to be duplicates of each other and choose to retain only one of them.

This work is part of a course CS4984/5984 Big Data Text Summarization. We were given a collection of 12,373 articles related to two shooting attacks that took place in Maryland to summarize. We followed the 10-step approach suggested by the course instructor, Dr. Edward Fox, to generate summaries where each step was more sophisticated than the previous in order to learn and develop in-depth knowledge of various NLP techniques and automatic text summarization. Along with the techniques involved in summarization, we learned the need for cleaning the collection and ways to remove noise. We also tried various classification techniques and tips to improve the accuracy of the classification techniques. By the end of this project, we were successfully able to generate a readable summary from the collection using the Pointer Generator deep learning technique.

9 Golden Standard Summary for Team 1

9.1 Introduction

We were also given the task to create the golden standard summary for Team 1 which handled the Hurricane Harvey corpus. For the manual summarization process, each team member worked on a topic in order to ensure everyone contributes evenly towards this task. This is the final version of the golden standard summary for Team 1.

9.2 Results

Hurricane Harvey initially developed in the Atlantic Ocean (11), to the east of the Windward Islands, making landfall in Barbados and Saint Vincent, and then moved into the Caribbean Sea (12). It achieved tropical storm status between August 17th and 19th 2017. It crossed the Yucatan Peninsula and then intensified over the Gulf of Mexico (13). It made landfall at San José Island near Port Aransas, Rockport, and Fulton along the Texas coast on Friday, August 25th (14). It tracked northwest through early Saturday, then slowly turned toward the east over the weekend. On August 28th, as a tropical storm, it moved back into the Gulf, causing torrential rainfall as it tracked east past Houston and Galveston (15). It made landfall again on Wednesday, August 30th, east of Port Arthur, Texas and also west of Cameron, Louisiana (16) It continued north and then northeast, past Lake Charles, largely dissipating as it moved into Kentucky, being absorbed into another extratropical system on September 3rd (17). Hurricane Harvey was categorized as a category 4 hurricane, which means that the maximum sustained wind was 130 - 156 mph (18). The highest wind speed recorded was 132 mph. Hurricane Harvey had a minimum central pressure of 937 mbar (19).

Hurricane Harvey caused catastrophic flooding and set a record for rainfall in the US when Nederland in Texas received over 60 inches (20). The death toll due to Hurricane Harvey was at least 107 people (19). The University of Wisconsin Space Science and Engineering Center referred the Hurricane as a one in a thousand-year flood event*. It caused more than 50 inches of rainfall in parts of Houston, the fourth largest city in the US (20). It also caused significant structural damage to sturdy buildings, wall failures and complete destruction of many mobile homes. Damage was greatly accentuated by large airborne projectiles, such as a cargo trailer hurled into the courthouse in Rockport (21). An estimated 300,000 structures and half a million cars were damaged or destroyed in Texas. Road transportation was also affected to a large extent with lots of Houston area roads being flooded (22). The Interstate 610 loop had accumulated about 2 feet of water. Many exits were closed along freeways and tollways, cutting off numerous neighborhoods (20). Houston airports and

the Port of Galveston were also affected. Around 704 flights were canceled at George Bush Intercontinental Airport and around 123 flights were canceled at William P. Hobby Airport (23). Many airport personnel along with hundreds of passengers at William P. Hobby Airport. 20,000 people were stuck at sea in four cruise ships at Galveston (23). Power outages were reported to cause four deaths of elderly people along with other damage such as disruption of a city wastewater treatment plant and the loss of fish at UT's Marine Science Institute Education Center (24). The total damage is estimated at 125 billion dollars, making Harvey one of the top-five most costly natural disasters in the US, and the second costliest hurricane (23).

Authorities had issued mandatory evacuation in several places where Hurricane Harvey was expected to hit. Harvey was a predictable storm with Genesis 12 days in advance of landfall. Officials warned about severe flooding on the 26th of August 2017. On August 27th, some authorities warned people to stay indoors (25). This was mainly to avoid people driving on flooded roads where many fatalities occur. About 4,500 inmates in the Ramsey, Terrell, and Stringfellow units in the community of Rosharon in Brazoria County were evacuated by buses to the facilities in East Texas. Additionally, around 200 hospital patients had also been evacuated from Corpus Christi and approximately 250,000 people were evacuated from their homes (26). Red Cross opened two emergency shelters in Houston to help people who had to be evacuated. These people were equipped with flood buckets, hygiene kits, blankets, tools, and equipment through volunteers from Cares Disaster Response Team (27). Dallas and Houston also opened their convention centers to shelter 5,000 people (28).

Various measures were taken to achieve restoration. The Public Assistance Program provided supplemental federal disaster grant assistance for debris removal, life-saving emergency protective measures, and the repair (29). The grant assistant also helped the replacement and restoration of disaster-damaged, publicly-owned facilities. It took about 18 days for the main electric wires and utilities in the Houston metropolitan area to resume power. To help people cope with the disaster damages, President Trump directed federal aid toward the state's recovery efforts in affected areas (30). 450,000 people were likely to seek federal aid, which includes financial help with rent, repairs, and lost the property. Chevron Corporation announced a 1-million dollar contribution to the American Red Cross (31). The Cantor Fitzgerald Relief Fund held the Hurricane Harvey Family Relief distribution event, giving out a total of approximately 5 million dollars to those families hardest hit by Hurricane Harvey (32). ConPRmetidos, a Puerto Rican organization focusing on public-private partnerships, raised 150,000 dollars for relief efforts. Many organizations and churches accepted donations for the cause of hurricane relief (33).

10 Acknowledgement

We would like to extend our gratitude to Dr. Edward Fox. He has always been so patient and encouraging during our learning phase. He also provided us with a large number of resources that we were able to use in order to produce the desired summaries and result. We would also like to thank our TA Liuqing Li who was always prompt in addressing to our queries however trivial it might have been. We are also grateful to Chreston Miller from Team 7 for helping us with preprocessing our data for pointer-generator. We would also like to thank Team 14 for providing us with the golden standard summary.

We would also like to thank the Virginia Tech Writing Center for helping us in improving the Gold Standard summary for Team 1.

We would like to acknowledge NSF-funded Global Event and Trend Archive Research (GETAR) project, IIS-1619028, which provided the given corpora.

Bibliography

[1] M. Allahyari, S. A. Pouriyeh, M. Assefi, S. Safaei, E. D. Trippe, J. B. Gutierrez, and K. Kochut, "Text summarization techniques: A brief survey." *CoRR*, vol. abs/1707.02268, 2017. [Online]. Available: http://dblp.uni-trier.de/db/journals/corr/corr1707.html#AllahyariPASTGK17

[2] E. K. Steven Bird and E. Loper, *Natural Language Processing with Python*. [Online]. Available: https://www.nltk.org/book/

[3] "Naive bayes." [Online]. Available: https://mahout.apache.org/users/classification/bayesian.html

[4] D. M. Blei, A. Y. Ng, and M. I. Jordan, "BleiNgJordan2003.pdf," vol. 3, pp. 993–1022, 2003.

[5] "K-means clustering." [Online]. Available: https://mahout.apache.org/users/clustering/k-means-clustering.html

[6] A. See, P. J. Liu, and C. D. Manning, "Get To The Point: Summarization with Pointer-Generator Networks," 2017. [Online]. Available: http://arxiv.org/abs/1704.04368

[7] "Get to the point: Summarization with pointer-generator networks." [Online]. Available: https://github.com/chmille3/pointer-generator

[8] J. D. M. Rennie, L. Shih, J. Teevan, and D. R. Karger, "Icml03-Nb.Pdf," *People.Csail.Mit.Edu*, no. 1973, 2003. [Online]. Available: http://people.csail.mit.edu/jrennie/papers/icml03-nb.pdf{\%}5Cnpapers2://publication/uuid/B3C25F39-7B42-4ADB-A47C-42595E1D526B

[9] Y.-n. Chen, W. Y. Wang, and A. I. Rudnicky, "for Spoken Dialogue Systems Using Frame-Semantic Parsing," *Proc. Automatic Speech Recognition and Understanding Workshop*, pp. 120–125, 2013.

[10] T. Kelvin. (2018) Lucene query syntax. [Online]. Available: http://www.solrtutorial.com/solr-query-syntax.html

[11] "2017 atlantic hurricane season forecasts, data, maps," May 2018. [Online]. Available: http://www.artemis.bm/blog/2017-atlantic-hurricane-season/

[12] 2018. [Online]. Available: http://www.kintera.org/faf/home/default.asp?ievent=1176065&lis=1&kntae1176065=DCBEC0AEAD73447DA7EAEFB2FF62326E&login=t

[13] "Analyzing harvey's path to destruction," Sep 2017. [Online]. Available: http://www.thepacer.net/analyzing-harveys-path-to-destruction/

[14] "Harvey creating life-threatening flooding in southeastern texas, early damage estimate up to $2 billion," Aug 2017. [Online]. Available: https://www.canadianunderwriter.ca/associations/harvey-creating-life-threatening-flooding-southeastern-texas-early-damage-estimate-high-us2-billion-1004119666/

[15] "Thousands stuck on cruise ships after harvey, 2017." [Online]. Available: https://amp.cnn.com/cnn/2017/08/26/us/cruise-ships-harvey/index.html

[16] V. Sharma, "Houston remains submerged as hurricane harvey makes second landfall; storm irma building up," Aug 2017. [Online]. Available: https://www.ibtimes.co.in/houston-remains-submerged-hurricane-harvey-makes-second-landfall-storm-irma-building-740367

[17] E. S. Blake and D. A. Zelinsky, "Hurricane Harvey," *National Hurricane Center Tropical Cyclone Report*, no. January, pp. 1–76, 2018.

[18] R. Andrews, "'unprecedented' hurricane harvey is 'one of the worst disasters in us history'," Aug 2018. [Online]. Available: https://www.iflscience.com/environment/unprecedented-hurricane-harvey-worst-disasters-us-history/

[19] 2018. [Online]. Available: https://en.wikipedia.org/wiki/Hurricane_Harvey

[20] K. Philips, "Texas flood disaster: Harvey has unloaded 9 trillion gallons of water," Aug 2017. [Online]. Available: https://www.washingtonpost.com/news/capital-weather-gang/wp/2017/08/27/texas-flood-disaster-harvey-has-unloaded-9-trillion-tons-of-water/?utm_term=.2bd31895e748

[21] "Seniors treated after roof collapse in rockport, texas." [Online]. Available: https://www.foxnews.com/us/seniors-treated-after-roof-collapse-in-rockport-texas

[22] K. Sullivan, "Harvey leaving record rainfall, at least 22 deaths behind in houston," Aug 2017. [Online]. Available: https://www.chicagotribune.com/g00/news/nationworld/ct-hurricane-harvey-flooding-houston-20170829-story.html?i10c.encReferrer=aHR0cHM6Ly9lbi53aWtpcGVkaWEub3JnLw==&i10c.ua=1&i10c.dv=18

[23] 2018. [Online]. Available: https://en.wikipedia.org/wiki/Hurricane_Harvey#cite_note-WP_Aug29-29

[24] A. Stuckey, "Ut marine science institute at port aransas still in shambles after hurricane harvey," Dec 2017. [Online]. Available: https://www.houstonchronicle.com/news/houston-texas/houston/article/UT-s-Marine-Science-Institute-still-in-shambles-12456377.php

[25] T. Acosta, "Mccomb: No mandatory evacuation order during hurricane harvey was 'right call'," Oct 2017. [Online]. Available: https://www.caller.com/story/news/local/2017/10/12/mccomb-no-mandatory-evacuation-order-during-hurricane-harvey-right-call/758159001/

[26] "A makeshift army, marching on adrenaline," Sep 2017. [Online]. Available: http://wapo.st/harvey-first-responders?tid=ss_tw

[27] ArunasNepalRelief, "Helping fellow texans and others! hurricane harvey relief! – arunasnepalrelief, inc." Sep 2017. [Online]. Available: https://arunasnepalrelief.org/helping-fellow-texans-others-hurricane-harvey-relief/

[28] "Houston's convention center turned into shelter for harvey flood victims." [Online]. Available: https://www.foxnews.com/us/houstons-convention-center-turned-into-shelter-for-harvey-flood-victims

[29] "Update on department of education response to hurricane harvey and hurricane irma," Sep 2017. [Online]. Available: https://www.ed.gov/news/press-releases/update-department-education-response-hurricane-harvey-and-hurricane-irma?utm_content=&utm_medium=email&utm_name=&utm_source=govdelivery&utm_term=

[30] J. Diamond and S. Tatum, "Trump makes disaster declaration for hurricane harvey," Aug 2017. [Online]. Available: https://www.cnn.com/2017/08/25/politics/trump-harvey-declaration/index.html

[31] [Online]. Available: http://www.riskmanagementmonitor.com/hurricane-harvey-hits-texas-with-up-to-30-billion-in-damages/?utm_source=feedburner&utm_medium=email&utm_campaign=Feed:RiskManagementMonitor(RiskManagementMonitor)

[32] "Hurricane harvey." [Online]. Available: https://www.cantorrelief.org/project/hurricane-harvey/

[33] "Couche-tard/circle k launches a nationwide campaign to support texas communities and its employees," Aug 2017. [Online]. Available: https://www.businesswire.com/news/home/20170830006257/en/Couche-TardCircle-Launches-Nationwide-Campaign-Support-Texas-Communities