# Exploratory Text Analysis Final

### Annapolis Police Department Daily Reports

Ben Ballard (bkq5nt@virginia.edu)

DS 5001

Figure 1: Example Report

### **A screenshot of a car theft**1. Introduction

I recently moved to Annapolis MD. The police post daily reports to a city website of press releases regarding crime and public service announcements. I wrote a web scrapper getting the report texts into a format for Exploratory Text Analysis.

**Patterns in reporting:** The chart below highlights that there are both report dates and incident dates, and they correlate but the incidents live at a higher level. The winter months have the least reporting Dec, Jan, Feb.

Figure 2: Report Seasonality: Monthly and Yearly

A graph of different colored lines

Description automatically generated with medium confidence

### 2. Methodology

**Text processing pipeline overview:** I have been able to pull approximately 7 years of reports - over 2000 entries. Scraping data of Title, Date Published, URL, and the text contained within the link, the reports. The reports typically contain information on (potential) crimes or public disturbances, these include an Incident ID. I used regular expression to extract these Incident descriptions with an Incident\_ID, seeing this as the MDU. My code also segments other sections from reports, i.e. hiring notices, public programs, etc, but this wasn’t the focus of my efforts.

**Document structure normalization (F2):** The raw text required further processing, incident reports were mined for incident dates, which could vary from the report date. Each incident was then normalized into a standard format containing fields for date, location, and description. This structured data was loaded into tables following the Standard Text Analytic Data Model (STADM), creating LIBRARY, TOKEN, and VOCAB tables.

The KDE plots below show that most reports dates are within 1 day of the incident date. There are occasionally large differences in dates which could be event planning or perhaps data quality errors. Incidents are on average 163 words.

**A screenshot of a graph

Description automatically generated**

**NLP feature extraction (F3):** Each document was enriched with linguistic annotations including parts of speech, named entities (particularly locations and times), and syntactic dependencies. These features were added to the TOKEN table. The VOCAB table was enhanced with document frequencies and term statistics. Sentiment analysis using VADER provided emotional valence scores for both individual terms and complete documents.

**Vector representations (F4):** TF-IDF vectorization was implemented using scikit-learn, treating each incident as a document. Terms appearing in >95% of incidents or in only one incident were filtered out. A document-term matrix was created (shape: (1193, 7886)) capturing the TF-IDF scores for each term across all incidents. This matrix was converted to sparse format for efficient processing and served as input for subsequent analytical models.

### 3. Analysis & Findings

**3.1 Document-Level Patterns: PCA:** The first 4 principal components make around 15% of the total variance of the total variance in the document term matrix. The biggest explanation of variance was component 1: vehicle: 0.39, victim: 0.303, stolen: 0.268, suspect: 0.256, contact: 0.242, block: 0.190, theft: 0.163, bicycle: 0.144, burglary: 0.137, home: 0.134**A graph with a red line

Description automatically generatedA diagram of blue dots

Description automatically generated** The scatter plot shows the first two PCs separated the documents into a few topical clusters, even if loosely.

**T-SNE:** I used t-SNE for dimensionality reduction to visualize semantic clusters in a word embedding model. Word vectors were extracted, normalized, and projected to 2D space. K-Means identified coherent thematic groups, labeled below.

A screen shot of a diagram

Description automatically generated

Some clusters that stand out include descriptions around 1.) community tips 2.) arrest descriptions. But admittedly these topics blend together to me – so could use refining.

**Hierarchical Cluster Diagrams:** This dendrogram shows a hierarchical clustering of word embeddings, revealing semantic groupings. The most notable feature is the split into two main clusters, indicating a fundamental difference in the vocabulary. The y axis represents semantic distance - the higher the connecting lines between clusters, the more different their meanings are. We can see how words group together first tight clusters at the bottom, then gradually merge into larger semantic categories, creating a comprehensive map of how these words relate to each other in meaning.

**A screen shot of a graph

Description automatically generated**

### 3.2 Language & Content Analysis

**Sentiment**: Not surprisingly most of the documents are negative. Using the Vader methodology, 1400 of the documents are negative with 490 positives The most negative terms are as expected, kill, rape, murder, suicide, fatality. I looked up the most negative incident on the document level, and I’ll spare you the details.

For the most positive entry, it started very negative which surprised me, but it then became much more positive as the incident bled into what was essentially an announcement for people to shop in downtown Annapolis. There’s clearly more work to be done on cleaning the data. Looking up positive entries could be a way to evaluate the data quality and look for text that needs further cleaning.

The winter months have slightly more positive sentiment, or perhaps just less negative than in the summer, when there is a higher number of incidents.

A graph of a bar graph

Description automatically generated with medium confidence

A graph of a number of data

Description automatically generated with medium confidence

I looked at seasonality over time of sentiment polarity and subjectivity, I was curious if there was a kind of step change due to covid, but I wasn’t able to identify a change.

A graph with blue and orange lines

Description automatically generated

### 3.3 Topic Analysis

**Latent Dirichlet Allocation**: The LDA topic modeling revealed five categories, offering insights into the types of communication and incidents documented. After tuning, the most coherent categorization emerged with topics that capture different aspects of the police-community interaction. The largest category (1), is on-going routine updates for getting in touch with the police department. Category 2 seems to focus on specific short term events the public should be aware of, i.e. traffic rerouting. Topic 3 and 4 seem to describe incidents focusing on direct criminal activity (terms like "victim," "suspect," "robbery") and Topic 4 centers more on vehicle-related crimes (terms like "vehicle," "theft," and "stolen"). The fifth and smallest category appears to capture oddly formatted elements from the report, suggesting data quality from the web scraping.

A screenshot of a computer

Description automatically generated

### 4. Conclusions

**Limitations:** I’ve learned that a huge percentage of the work for Text projects are formatting and processing. While modeling is always considered the fun part, I’ve found a huge amount of iteration is required here, significantly more than tabular data. While this could be because I’m new to NLP work, I see some structural differences in working with language that makes it even more iterative.

**Future Work:** I would like to do experimentation with Bert models to provide additional contextual understanding. I’d like to publish future versions of this analysis and publish to the community. Please provide any recommendations on analysis.

**Notes on Data Cleaning:** Web scraping is a tricky business. Even though I have the full text, I would keep finding interesting characters or long tokens where the spaces were removed. I found I had some very long strings that become huge tokens. I had one that was over 200 characters. Perhaps in the scrapping I lost spaces, I’d have to figure out where this came from. I wrote a crude rule to cut off tokens larger than 50. I found interesting terms like “wwwnavysportscomtickets11navyticketsbaskblhtml”. This is likely web links artifacts. They could provide value, but for now I just decided to cut it out of my pipeline. **Domain Stop Words:** I have a better understanding now of how this is an iterative process, because there are terms that I’m not prepared to just remove from my pipeline, but for certain questions they may need to be removed.