# Exploratory Text Analysis of Annapolis Police Department Reports

Ben Ballard

DS 5001

12/10/2024

Figure 1: Example Report

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

I recently moved to Annapolis MD. I’ve wanted to apply my data science course work to have a better understanding of my community. I know the police post daily reports to their website of notices that need to make it to the public, whether regarding crime or just PSAs. I refocused my energy on the web scrapper and getting the text into a format for ETA.

**Patterns in reporting:** The chart below highlights that are 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.

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 led me to meta data of Title, Date Published, URL, and the document Text contained within the link. The report texts typically contain incidents if they have information on (potential) crimes or disturbances, these include descriptions and 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**

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.

**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.

This matches your implementation which:

1. Groups by incident\_id
2. Uses TfidfVectorizer with max\_df=0.95 and min\_df=2
3. Creates a document-term matrix for F5

**Analytical models applied**

### ## 3. Analysis & Findings

### 3.1 Document-Level Patterns

**PCA**

The first 4 principle 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**

A screen shot of a computer screen

Description automatically generatedI used t-SNE for dimensionality reduction to visualize semantic clusters in a word embedding model. Word vectors were extracted, normalized, and projected into 2D space. K-Means then clustering identified coherent thematic groups, which were labeled and annotated on the visualization.

**-Hierarchical cluster diagrams**

**▪ Heatmaps showing correlations**

**▪ Scatter plots**

**▪ KDE plots**

**▪ Dispersion plots**

- Distribution of incident types:

- [Visualization: Time series or seasonal patterns]

### ### 3.2 Language & Content Analysis

- Key terminology and phraseology patterns

- Named entity analysis (locations, types of incidents)

- **Sentiment** patterns across reports

Not surprisingly most of the documents are negative. Using the Vader methodology, 1900 of the documents are negative, with 3 positive. The most negative terms are most negative kill, murder, rape, kill and suicide. I looked up the most negative incident and I’ll spare you the details.

I looked for the most positive entry. It started negative… and then became much most positive when the incident bled into what was essentially an announcement for people to shop in downtown Annapolis. So 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.

A graph of a number of data

Description automatically generated with medium confidence

A graph with blue and orange lines

Description automatically generated

- [Visualization: Term importance or topic distribution]

### ### 3.3 Topic Analysis

- Discovered themes or categories

- Changes in topic distribution over time

- [Visualization: Topic model results]

### ## 4. Conclusions

- Key insights about police reporting patterns

- Potential implications for public safety communication

- Limitations and future work

**Notes on Data Cleaning:**

Web scraping is a dangerous business. Even though I have the text, I would keep finding interesting characters. I found that 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.

So I wrote a crude rule to cut off tokens larger than 50.

But then when looking at some times, I found interesting terms like  
“runsignupcomracerunforthedogsinblue5krun1kwalk” or “wwwnavysportscomtickets11navyticketsbaskblhtml”. I think these were likely web links of some sort. And they could provide some value, but for now I just decided to cut it out of my pipeline.A graph of a distribution of token length

Description automatically generated

**Domain Stop Words:**This is challenge, as there a lot of terms that do occur a lot. 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.