Opioid Overdose Data Analysis

Technical advisor: Amarnath Gupta
Domain expert: Annick Borquez
Team members: Kevin Liu, Jimmy Quach, Fangzhou Hu, Yang You

Abstract

Drug use epidemics are fast-evolving and our public health response often lags far behind. The project gathers opioid outbreak data from multiple news websites in the United States, uses the SDSC's AWESOME platform to organize all data and creates a geospatial topic model construction framework by automatic analysis of variable co-variation. The articles and their metadata are displayed through an interactive map visualization. There is a correlation found between the data that is present within the map with the filter: "opioid" and the opioid map produced from the NIH, which allows the scientists to further investigate and perform a limited degree of what-if analysis.

Introduction and Question Formulation

Currently within the United States as well as other locations around the world, the use of drugs, whether it be their actual consumption or transportation, has continually increased with the advent of technology facilitating further usage and abuse. While drug use has continued to run rampant, the response by the government and other law enforcing officials has been more reactive than proactive. Thus, if there was some method or framework that could be able to determine trends or provide additional information of where an outbreak could occur, officials could be able to stymie and combat drug outbreaks rather than reacting and dealing with the aftermath after it's spread and affected numerous people.

Our problem ends up being, how can we develop something to assist personnel and researchers about this subject and allow them to take a proactive approach. The overall data science problem becomes quite extensive since there needs to be a focus of the initial data that can be used and analyzed to make these assumptions of outbreak. It would be quite difficult to have sensors or other factors of what entails a drug outbreak. Drug outbreaks do not necessarily have quantitative indications that can be analyzed in this way to ensure an outbreak is occurring. While the above is difficult, there is a data source that could be leveraged that is constantly updating and can provide an indication of an occurring drug outbreak, newspaper articles. By leveraging and analyzing newspaper articles, a platform can be situated on top of the analysis and extracted metadata from the newspaper articles to produce indications of trends that an outbreak is occurring.

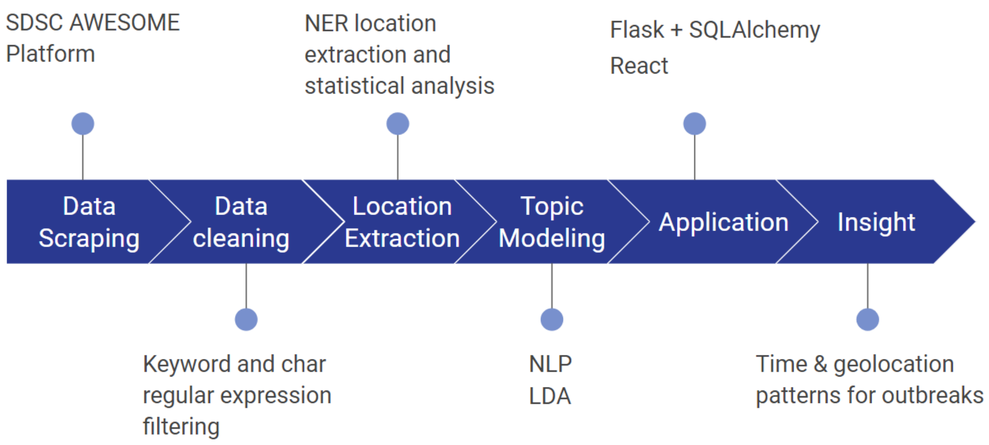
Team Role and Responsibilities

In terms of producing the above solution, there requires a few different roles to accomplish this. There needs to be a person allocated to performing the newspaper extraction and initial filtering and data cleaning of said articles which may include lemmatizing. There needs to be a person allocated to perform the topic modeling to indicate which articles are of interest as well as performing the location extraction to determine where each of the articles are located. Another person needs to be able to develop the api interface to interact with our data store and finally another person to create the web application where end users can interface with. Ultimately we had a similar team structure, where 2 people were involved in the more data-esque tasks of ETL and modeling to produce the relevant metadata and the other 2 people primarily worked on the engineering, developing the API as well as the application.

Data Acquisition

Our primary data acquisition occurs with obtaining newspaper articles. This primarily involves the web scraping and traversal of articles and feeding them towards a local data store. This step was accomplished previously due to our collaboration with the San Diego Supercomputer Center and their Awesome Platform. The Awesome Platform periodically scraped and inserted general newspapers that could be later used for downstream analysis. The data that they are pulling is rich, due to there not being a single source of where they are getting the articles from, rather from numerous sources. This lowers the subjectivity barrier of a drug outbreak that could happen due to the sheer number of potential articles that could be used to classify a drug outbreak.

The article data present within the Awesome Platform is widely varied in terms of size. There are initial metadata columns representing information for each article such as author and article date as well as the full text information for each article. With the full text being captured, each article can widely vary in terms of the amount of storage it occupies due to some articles being lengthier than others as well as the amount of metadata that gets extracted from each newspaper article. .


SDSC AWESOME Platform
NER location extraction and statistical analysis
Flask + SQLAlchemy
React
Data Scraping
Data cleaning
Location Extraction
Topic Modeling
Application
Insight
Keyword and char regular expression filtering
NLP
LDA
Time & geolocation patterns for outbreaks

Data Preparation

Since a breadth approach was taken to obtain newspaper articles, there was extensive data preparation that was needed to clean and filter the articles for future analysis. Firstly, the article scraping process to obtain newspapers is not perfect, resulting in some articles just consisting of links or ads rather than actual news content and thus had to be filtered out. Furthermore, as mentioned above, if the newspaper scrape ends up successful and the raw text is obtained, the raw text may still have to be cleaned because there still may be affiliated links to other articles or ads present within the text. Also, the raw text may not be in English either and the platform that is being developed is primarily aimed towards an English-speaking audience and thus those as well, had to be filtered as well.

With regards to the actual transformation of the data, conditionals were applied to the data such as the ones mentioned above to produce a subset of the data. The raw text data was also further lemmatized and stop words removed to ensure that the words remaining would have a higher relevance to the overall article. These newly synthesized corpuses were then checked against a list of words that we believed were valuable and relevant. If the corpuses did not contain these specific words, that specific article was essentially removed from further analysis.

These methods were chosen because there was a desire to reduce the amount of analytical processing of the articles as well as have a robust article set to perform analysis on. Specifically, these features and methods were chosen because they were fairly lightweight in terms of their actual processing, the article has this specific element or it does not as well as their ability to remove many articles at once. Ultimately this was done because articles that were irrelevant, for example an article on cooking, would have zero relevance to drug outbreaks and may ultimately dilute the fidelity of the end solution.

By performing the pre-processing methods, the overall data set that we obtained had a higher fidelity and relevance to our main goals compared to the wide varying multitude of articles that the Awesome Platform initially contained. This allows the downstream analysis to be performed quicker and the results to have more meaning and weight as the data set is highly relevant to the overall goal rather than something that was once ambiguous.

Analysis Methods

The analysis method is mainly driven by our research interest. The application aims on providing location insights about news information related to opioid or other drug related issues. To achieve this goal, we need first figure out which news corpus are related to our research interest. Among these articles, we need to extract the location information from the raw news corpus. The topic modeling and location extraction are both two equally important tasks for our analysis modeling.

As this is an unsupervised learning task, we start with Latent Dirichlet Allocation topic modeling. However, after several rounds of tuning this modeling result is not good enough (reference to Raw Dataset LDA model in table attached below). Among the 10 topics we found only two of them are related to health care issues. After reading some articles, we started to realize that the web scraping data quality is not as good as we expected. For example, some article corpus starts or ends with some strange sentences, which is unrelated to the actual content of the article. After tracking down the actual URL, we realized these sentences are part of the web advertisement element or navigation element instead of the main body of the articles. For advertisement slogans, it does not harm the data quality too much as the content from web advertisement is usually unrelated to our interesting topic. However, some content from navigation channels, like title links to other articles, severely damage the data quality. We have seen cases, where the main body of the article content talks nothing about the opioid related stuff, but at the end of the articles there are three other articles' titles containing keywords opioid or addiction, as some recommendation reading material. These opioid related sentences affected our LDA topic model quality and LDA mistakenly marks these unrelated articles under "opioid" topic. In order to perform better topic modeling, we needed to have cleaner data.

To solve this problem, we first introduced a new concept called keywords. Keywords are nouns that have the 10 highest occurring frequency within each article. With help from our domain expert, we defined a target keywords pool (attached below) and then we selected articles which have at least 2 keywords hits within the target keywords pool (attached below).

Keywords Pool: opioid, opioids, overdose, overdoses, drug, drugs, fentanyl, drug, health, harms, overdose, deaths, fatal, overdoses, hiv, hcv, tb, endocarditis, infectious, diseases, mental, , emergency, room, treatment, opiate, agonist, therapy, health, system, policies, disorders, mandating, prescription, drug, monitoring, programs, naloxone, samaritan, enforcement, drug, seizures, police, crackdowns, drug, busting, property, violent, crime, associated, with, drug, use, drug_related, arrests, incarcerations, drug, cartels, drug, markets, , fda, approvals, trends, self_medication, pain, , recreational, club, , disorders, dependence, prescription

After the filtering process discussed above, we obtained a smaller data set, which is ⅛ of the original size. With this new data set, LDA modeling is applied again and the resulting topics are attached below. From the comparison among the raw dataset LDA model, filtered dataset model and expert defined topics, filtering does help us better define a topic model, which is closer to the target topic list defined by field experts.

Raw Dataset LDA model	Filtered Dataset Defined	Expert Defined Topics (goal)
		● Drug use related health harms

Topic #0: food like coffee restaurant new chicken sugar best make good eat day chef water bar Topic #1: said photo house spending budget border 2018 government orange scng billion trump democrats immigration senate Topic #2: people like time said use think dont new facebook social thats make way know research Topic #3: like life story time times world new book people way said los angeles years family Topic #4: new star click instagram actress dress black look getty showed ap photos like left stars Topic #5: trump said president people new state white house health states percent government trumps tax years Topic #6: like said know im mr think people time dont la going thats really told got Topic #7: pm game games new company google apple app free players video million year st chico Topic #8: said people treatment years health addiction homeless county says care help like time family children Topic #9: said drug opioid drugs state marijuana police medical opioids new according federal people law health	Topic 0 : company, industry, drug, investigation, manufacturer, state, case, former, attorney, pharmaceutical Topic 1 : fentanyl, agent, machine, package, seizure, authority, synthetic, pound, chinese, lab Topic 2 : prescription, doctor, pill, patient, painkiller, pharmacy, medication, oxycodone, opioid, physician Topic 3 : public, emergency, federal, trump, law, crisis, drug, national, government, administration Topic 4 : treatment, addiction, recovery, center, addict, facility, bed, program, residential, people Topic 5 : crime, drug, prison, police, court, jail, prosecutor, case, charge, dealer Topic 6 : product, computer, market, device, psilocybin, generic, implant, fish, plant, potential Topic 7 : drug, opioid, death, year, overdose, heroin, people, epidemic, fentanyl, prescription Topic 8 : health, program, state, care, service, insurance, people, medical, year, also Topic 9 : patient, pain, opioid, drug, medication, study, doctor, medical, also, people Topic 10 : mental, percent, rate, health, child, adult, woman, baby, age, man Topic 11 : people, year, time, day, family, home, drug, life, back, even	○ Overdose deaths and non-fatal overdoses ○ HIV, HCV, TB, endocarditis and other infectious diseases ○ Mental health harms associated with drug use ○ Emergency room ● Drug use treatment (e.g. opiate agonist therapy) ● Health system policies related to drug use disorders (e.g. mandating prescription drug monitoring programs or naloxone laws Samaritan laws) ● Drug-related law enforcement ○ Drug seizures/ police crackdowns/drug busting ○ Property/violent crime associated with drug use ○ Drug-related arrests, incarcerations ○ Drug cartels/drug markets (This could eventually be its own category but trying to keep it as concise as possible at this stage) ○ Drug laws/policies, including FDA approvals ● Drug use trends ○ Self-medication ○ Pain management ○ Recreational use/club drugs ○ Drug use disorders/dependence ○ Drug prescription

To extract the location information from the news corpus we applied the Named Entity Recognizer (NER) method on labeling each word vector in the news corpus. Based on the word vector labeling data, we used the following algorithm to define the one location entity for the corresponding article.

```
1. remove nation names and other words like "city”, “road" from nlp result
2. if no location names left after 1:
    output empty
3. calculate first appearance and frequency for each location
4. if a location name is the first word of the article OR highest frequency location name is also closer to beginning of the article than other location names:
    output this location
else:
    output TBD
```

Naturally, the goal is to minimize the TBD output and improve the accuracy of the result, human reading is expected to shed light on both. After calculating the location information for all our interested articles, we spent some time manually extracting location information from these articles. So far, we have labeled 233 of them, among these articles, our location extraction algorithm achieved 66.2% of accuracy compared to human labeled location.

With a solid topic model, we established a processing pipeline as a daily cron job to refresh the modeling on a moving time window. As a result of the topic modeling processing, we map each news article with human readable topic labels that we can then present in the mapping application. Parallelly, we applied the location extraction algorithm defined above to map each article with a location entity. All the mapped information will be stored in the data structure listed below in the central database.

1. TopicDoc: a table stores articles relevant to our research interest and their topic ids calculation from LDA topic modeling.
2. OpioidDocLoc: a table stores location information for interested articles.
3. Opioid_label: A table stores topic label information provided by data experts.
4. Opioid_topic_label: A table stores relationships on how topic id can be mapped to human readable labels.

Findings and Reportings

Ultimately our goal was to provide an experience to end users allowing them to explore the newspaper articles within the Awesome platform and use them for their further analysis, specifically in this case, in the domain of opioid outbreaks. This goal was achieved through the combination of the data processing and analysis done to the relevant newspaper articles as well as through the creation of an API and front end application that would allow end users to consume the information programmatically or through an interface.

On the application side, the API has numerous routes that reveal information that has been analyzed and stored. The web application uses the API to then surface said information via a varying number of dashboards. Some of the features and visualizations that are present within the web application are a choropleth based map that has a variety of filters and searching options to allow end users the ability to filter through a wide range of articles to determine specific moments in time or subsets of the overall analyzed data. The filters and search options include filtering on time, keywords, topic, and location. Based on the use cases and personas we utilized, these seemed to be the most relevant options and features to search on. There are also other various features such as the ability to flag specific articles for further analysis and bar graphs that describe the distribution of some of the metadata elements present within the selected articles.

In terms of the fidelity of the analytical results, they seem to be quite fruitful. While the results may be difficult to verify since the determination of a drug outbreak is a subjective notion, we have seen a correlation between outbreaks and data present within the NIH's own visualization maps. While further validation would be necessary and a deeper diving into the articles themselves, it nevertheless does show a positive outcome that the analysis and application has some potential weight of usage to it.

Solution Architecture, Performance and Evaluation

Our architecture and product consists of 4 major portions: the AWESOME platform that is acting as our data store, the analytical scripts that we utilize to perform ETL processes as well as to perform the topic modeling and location extraction, the api layer that connects directly to the data store where users can run REST commands directly to obtain data, and the web application where users can explore the data through the visualizations and filtering that is provided on that level.

The analytical scripts are robust because of how they are executed and the non-reliance the application layer has on it. The scripts are run on a CRON basis (per day) and apply the scripts to the newly inserted data within the AWESOME platform. While the scripts are running, data can still be read from the data store and the application and api can still be used. Although the scripts may take a significant amount of time due to the overall inserted day, basic benchmarking has been done that the daily ingestion rate of the platform and the average of that does not exceed the time limit imposed on the analytical scripts. Although there could be days where the ingestion could be extremely large, the current script execution time is significantly less than a day and if anything, the cron jobs can be executed one after another or simply reapplied if there happens to be an error or anything similar.

On the application side, we initially decided to use Docker, a type of containerization technology, early on to be able to deploy and handle the application in variable environments.

We were initially planning on this due to our advisor mentioning the idea of being able to publicize the api / application as well as the idea of being able to host it into his own environment. We were unsure of what that target environment would be (if it was bare metal, virtual, cloud based, etc.) and by going this route of having everything containerized would reduce the necessary components to actually run the application to just be Docker which is widely used by numerous companies and supported by most if not all operating systems.

Furthermore, with the applications being containerized, it allows them to easily be load balanced by some form of proxy. The proxy can forward connections to various instances to balance compute resources across the various containers to continue to allow the application to be performant. Also, we have served the Python Flask api through a wsgi and have all the requests being served by various workers to bypass the global thread lock that Python has. This allows concurrent requests to the database and allows the application to be performant. In this situation, the application is limited to the performance of the database and the number of connections it can handle, however the management of the database is out of the scope for our project.

On the front-end portion, due to the sheer number of potential articles that can be retrieved from the api, we provide an initial default limiter to keep the browser performant. We allow users to increase the number of articles being retrieved, while providing them the proper details of the risk they are potentially taking by sending a large amount of data to the browser. Furthermore, to improve robustness, we added capabilities such as filtering the articles by specific metadata that we have decided upon such as predefined topic labels and keywords that have been extracted from the article. Rather than load all the articles onto the map, we use the article count per location to provide a color spectrum for the map. This means that regardless of how many articles get added to the articles tables, as this is just an aggregation, the front end will receive a similar amount of data from the backend.

Conclusions

Using analyzing newspaper articles for their respective locations and topics, there is some feasibility in terms of determining and predicting drug outbreaks and trends. The application provides a means to allow end users the ability to further filter for specific articles to enhance and clarify the overall trends. There are requirements for the data and topics to be continually updated to prevent staleness and ensure freshness since what can be considered a drug outbreak can constantly change and what may be relevant in the past may not be relevant now, but nevertheless, the platform has shown potential in being able to predict potential trends and provides a means for users to continue and perform additional research.

References

[Cities Geojson] https://catalog.data.gov/dataset/500-cities-city-boundaries-acd62

[Geojson for state and County] https://eric.clst.org/tech/usgeojson/

[Mapping of state abbreviation] https://github.com/TexasSwede/stateAbbreviations

[geojson simplifier] https://mapshaper.org/

Appendices

a) DSE MAS knowledge applied to this project: data integration, machine learning, python programming, database and visualization.

b) Link to the Library Archive for Reproducibility
Quach, Jimmy; Hu, Fangzhou; You, Yang; Liu, Kevin; Gupta, Amarnath; Borquez, Annick (2020). Opioid Overdose Data Analysis. In Data Science & Engineering Master of Advanced Study (DSE MAS) Capstone Projects. UC San Diego Library Digital Collections. https://doi.org/10.6075/J0V986KF