Measuring public sentiment of the Chicago Police Department

June 6, 2019

Abstract

Although public perception of the police can help unlock drivers and results for police effectiveness, it is extremely hard to measure. In this project, I attempted to classify whether tweets expressed positive, negative, or neutral sentiments of the police. To do this work, I used a random sample of about 1500 tweets from May 2020. MTurk workers coded these tweets as neutral, not relevant, positive, or negative. I used traditional classifiers (Naive Bayes and Logistic Regression) as well as neural net classifiers to predict tweet label in three ways. First, to predict four-way classification. Second, to predict opinion tweets from all tweets. Third, to detect negative opinions from all tweets. Although no classifier was able to accurately and precisely predict general classifications of tweets, many showed promise. Work with larger and more robust datasets could potentially increase predictive power of these models.

Contents

1 Introduction

Public perception of the police is incredibly important to police effectiveness and legitimacy but extremely difficult to measure. Public perception offers insight into how well a police department is functioning and may suggest adherence to tenets of procedural justice. Yet, compared to traditional performance metrics, metrics to evaluate public opinion are poorly defined and documented.

There is very little research currently available that measures real-world public sentiment of the police in US cities. As a result, this project serves to provide a proof-of-concept first and foremost that publicly-available data can be easily acquired and use in order to study common issues in policing.

1.1 Clearance rates

Homicide clearance rates, or what share of murders a police department "solves", are a key performance metric for police departments. Chicago has one of the lowest homicide clearance rates in the country, and only about 1 in 6 murders lead to arrest. Moreover, Chicago's clearance rate has steadily declined over the past ten years, from about 40% in 2000 down to under 20% in 2017. As a comparison, several police departments have markedly higher clearance rates. Over the past decade, Los Angeles has solved 51% of murders and New York has solved 61% of murders.

There are several potential reasons for the low clearance rate in Chicago, some of which suggest that non-traditional metrics of policing like procedural justice or public opinion may be re-

lated to traditional metrics. Police officers tend to cite the historically fraught relationship between the people and police, believing that someone who already views the police negatively because police seem inept may be less likely to cooperate with an investigation; more bluntly, many police officers lament a "no snitch" policy among victimized communities in Chicago. Evidence is conflicted: the National Crime Victimization Survey reports that these communities are no less likely to report crimes to the police, but a Cato Institute survey shows a race and education gap for crime reporting. There are also other viable explanations for Chicago's abysmal clearance rate, most notably that Chicago's police force has limited manpower per murder. Chicago has more murders than New York and Los Angeles combined, yet the police department (12,000 officers) is dwarved by New York's (36,000) and Los Angeles' (10,000).

1.2 Public opinion

Public opinion can also help measure procedural justice, or how police officers enforce laws. Although procedural justice is difficult to measure directly, past research has evaluated procedural justice through the lens of public opinion survey data.

Procedural justice is necessary for effective policing. A civilian who considers the law enforcement process fair and just is likely to consider any related consequences fair and just, too. Conversely, when civilians perceive lack of procedural justice, they are more likely to file complaints and view their police force as delegitimate. For example, one study of New York Police Department Stop, Question, and Frisk stops showed that civilians who believed their stop to be fair were less likely to file a complaint than those who believed their stop was unjust. Finally, a lack of procedural justice in just a few encouters can severely curtail public opinion of the police. Negative interactions with the police shape citizen perception up to fourteen times more strongly than positive ones.

Public perception of the police offers an additional metric to assess police performance. While hard metrics like clearance rates are easy to measure, assessing how the public feels towards the police is far more complex. Indeed, most work that tries to assess public sentiment uses survey-based or experimental research. More recent work has considered sentiments of tweets to assess public opinion of the police. As a caveat, public perception of the police is complicated and interacts with policing in myriad ways.

1.3 Research goals

The goal of this work is to assess the extent to which twitter data can be reliably used to evaluate public opinion of the police department. There is no dataset on police-related tweets that I could draw on, for example; very little by way of classifying tweets as police-related or not exists at present. This work then serves to primarily explore whether tweets can be reliably be categorized as "police relevant, positive", "police relevant, negative", "police relevant, neutral", or "not police relevant" with respect to public sentiment.

Why might such work be important? There are thousands of tweets about policing each and every day in the United States. Understanding which tweets reflect public sentiment (rather than are unrelated but use similar acronyms) provide a foundation for further research. More importantly, understanding what precise sentiment tweets express on a larger scale can better enable researchers to measure public sentiment of the police.

More concretely, this is simply the beginning. Once we can reliably predict how a tweet relates to policing, we can begin to assess public perception of the police. More specifically, I'm interested in assessing the extent to which public sentiment reflects traditional metrics of police effectiveness, where effectiveness here is roughly equivalent to clearance.

2 Past Work

2.1 Using twitter data to measure public sentiment towards the police

Although there has been limited work using data science techniques to study criminal justice, the Urban Institute used sentiment analysis for police-related tweets to measure how perception of the police changed due to the murder of Freddie Gray, using the following methods:

- Obtaining the data: researchers used a set of relevant tweets from 2014 and 2015 acquired through twitter.
- Processing the data: researchers removed mentions, hashtags, links, punctuation, and stop words from all tweets. They also used CoreNLP to tag tweets (e.g., to identify whether "cop" was a noun or a verb in each tweet).
- Learning models: researchers classified over 4,000 tweets manually to identify whether the tweet was positive, negative, neutral, or not applicable to their research for use in training and validation sets. They then used several types of models to predict the sentiment of new tweets and selected a gradient-boosted regression classifier as their model based on its accuracy (63%).
- Conclusions: researchers then used their newly labeled set of all tweets to assess the shift in public sentiment over time.

2.2 Using twitter data to connect public opinion with tweet sentiment

As a more general example, researchers at Carnegie Mellon University determined that public opinion surveys correlate to twitter sentiment on several key issues. They used twitter data specifically with two endgoals: to identify relevant tweets and to estimate sentiment (positive and negative) about a given topic. The specific methods used in this paper are less relevant to this work than the fact that twitter data alone was used to measure public sentiment.

3 Methods

Unlike my wiser classmates, I decided to create my own dataset using twitter data. Ultimately, data cleaning took an enormous effort – and I could continue to clean data forever and incrementally improve data quality with each run. Below, I walk through the decisions I made when creating my dataset, as well as some tradeoffs each decision posed.

3.1 Extracting tweets using the Twitter API

Tweets were extracted from the Twitter API using the following criteria:

- Tweets contained at least one of the following search terms: "Chicago Police", "CPD", "chicago police department", "second city cop", "chicago cop"
- Tweets were extracted in two rounds: one in early April that was overly dominated by the Jurnee Smollet case, and one at the middle of May.

Ultimately, tweets in the second set were used for modelling. Each dataset included over 15,000 tweets because the first dataset was weighted too heavily in terms of the Smollett case, an incident in which a famous actor allegedly falsely reported a hate crime. I did not want my dataset to be weighted heavily in terms of a single topic; rather, I wanted data to be more general.

As an aside: In retrospect, all news cycles have certain dominant stories. Were I to execute this project again, I would have used a sample of tweets from many time periods (say, 150/week over a period of a year) in order to ensure that my model was not simply learning how certain news events are labeled.

3.2 Preprocessing and labeling tweets

Preprocessing and labeling large amounts of text data posed a non-trivial challenge. I attempted several approaches, as listed below, in order to efficiently and accurately label the data. I used the following criteria to label tweets:

- *Not Relevant*: A tweet that my filters picked up, but was not actually about the police (e.g., it used a common acronym) was coded as not relevant
- *Neutral*: Neutral tweets discussed facts or headlines. Any news headline or simple fact (e.g., "Police Union Meeting Tomorrow at 6p!") was classified as neutral
- *Positive*: A tweet that expressed someone's positive view of the police was classified as positive
- *Negative*: A tweet that expressed someone's negative view of the police was classified as negative

To preprocess, I used the following pipeline:

- I stemmed all words
- I removed punctuation and irregular characters
- I removed citations for retweets (the handles that followed) and twitter handles where possible
- I removed hashtag symbols
- I attempted to translate emoji into English meanings

Note that depending on the way I handled data labeling, I was able to maintain different aspects of the tweets. I have flagged important compromises where I made them. Across the board, there were several data processing issues that I had to content with across methods:

- Emojis: in order to import all data, I needed to use "mac roman" encoding. This eliminated a lot of emoji data. As a result, emojis, which I had originally intended to translate to their corresponding words, could not be included across all datasets
- Tweet characters: For long tweets or tweets with retweet citations, not all of the data could be included in the text. Instead, these tweets became truncated.

3.2.1 Hand labeling

First, I tried to hand label a subset of tweets using these rules. Not only was this wildly inefficient, I also learned that I myself have bias. I didn't want my own bias to impact how I coded a tweet (for example, I noticed that I would code "bad" news stories as negative if someone retweeted them, but not "good" news stories). I also have experience working with the police, and so read significant subtext in tweets where subtext might not have been present (for example, if a tweet mentioned Second City Cop, a popular sarcastic blog that is pro-police, I often interpreted the tweet as sarcastic). Consequently, I decided that I would not hand label all tweets independently.

3.2.2 CoreNLP

The next thing I attempted was to use an out-of-the-box sentiment classifier to identify positive and negative tweets about the police. To do this, I configured an aws EC2 instance such that it was running Stanford's CoreNLP program on port 9000, and then submitted cleaned tweet text to the server for all unique tweets. Once I had my cleaned data, I examined the automated results.

This approach posed several challenges. First, and perhaps most importantly, it did not address the key issue (bias) from hand labeling. The out-of-the-box senitment labeling was problematic because it didn't fully account for context. As a result, many of the tweets were mischaracterized. Second of all, this approach was even more time intensive than labeling, since I had to set up the EC2 server,

As an example, the following tweet was classified as "negative". Because this tweet is a headline, it should have been classified as "neutral". This problem was pervasive throughout the dataset, and because tweets had already been preprocessed before getting sent to the CoreNLP

server, extremely difficult to fix. What's more, fixing the issue would be akin to hand-labeling tweets, which is what I wanted to avoid in the first place.

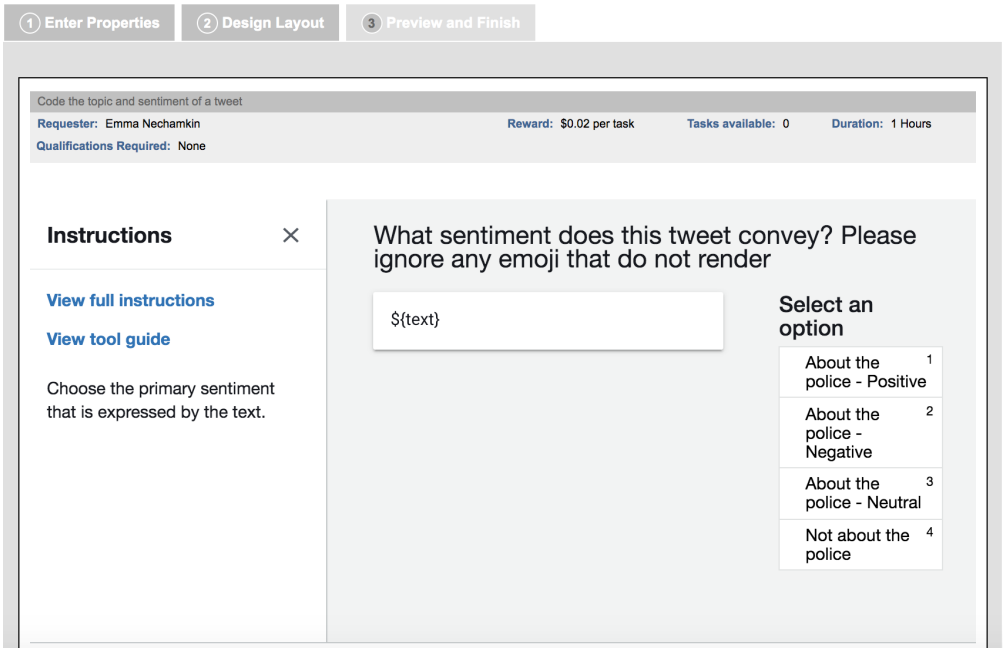
tweet: `chicago il tribune local chicago police searching two men briefly abducted 12yearold girl south side`

3.2.3 MTurk HIT task

The final method that I used to label - and the one I completed and eventually used - was to create a Human Intelligence Task on Amazon's MTurk to have workers label my text.

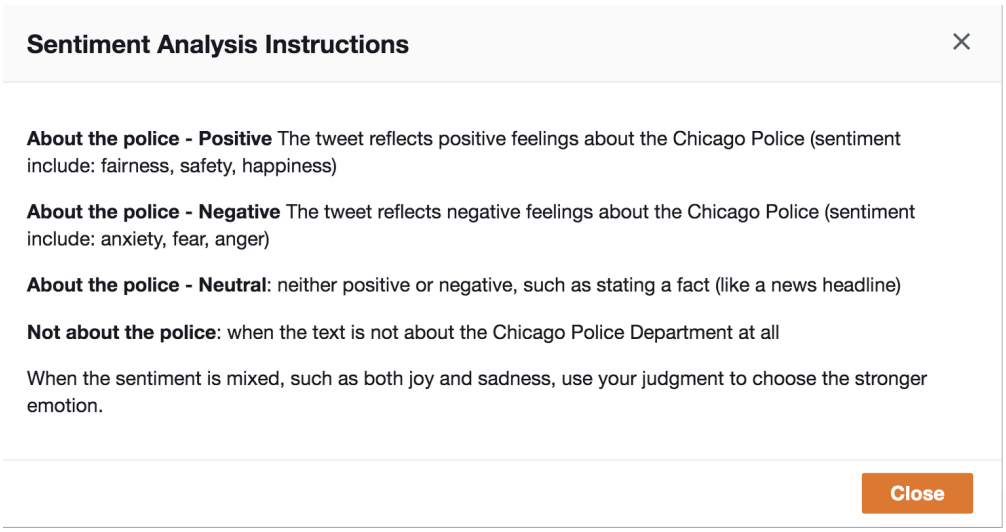
MTurk is a platform that connects requesters to workers. A requester is someone who wants a certain task completed by humans (e.g., labeling text). A worker gets paid per correct task completion. In this case, I submitted somewhat clean (but not stemmed) unique tweets to MTurk (a bit over 1500). The screens that workers saw are shown below.

Decision screen



This is the screen that all MTurk workers saw when classifying tweets

Additional instructions



These were the instructions that I provided for the task. Workers could refer to these instructions at any time

Once workers classified all of the tweets, I reviewed the work. I rejected over 800 tweets (about half of the original task load) due to blatant inaccuracies (e.g., facts as positively or negatively coded). In total, about 2400 tweets were coded by MTurk workers, of which I used about 1500 in model building.

This was my first time using MTurk. Were I to do it again, I would change several things:

- I would require worker qualifications to ensure that only good quality workers were coding my tweets. Reviewing the coding took an enormous amount of time. Not only did I have to review almost every single tweet, I also had to write explanations for every tweet that I rejected and correspond with workers about why their work was rejected.
- I would have each tweet coded by 3 high-quality workers as a way to flag irregularly coded tweets. I had to review every single tweet for this project, but had I had the means to ensure multiple eyes on each tweet, I surely would have, and would only have checked tweets with disagreement.
- I would have been clearer in my instructions by providing specific examples. I think many workers did not read my instructions carefully, and I believe my instructions were, in retrospect, vague. I would more carefully describe the task and also be sure to more clearly label different options for each tweet.

3.3 Final dataset

Across all data, the tweets were classified as:

Label	Count	Percent
Not relevant	439	28.6%

Label	Count	Percent
Neutral	541	35.2%
Negative	287	18.7%
Positive	270	17.6%

The dataset was split 50/50 into training and testing data. For the neural networks, the validation set was taken by random from the training set. For the more traditional models, there was no validation, as I didn't tune hyperparameters. In future work, once I have improved these models, I will also include validation.

My priority, however, with this run of the model was to increase my training set as much as possible in order to enable me to have as robust as possible data. This ended up being the right choice – my models became less precise as I decreased the training set size – even if untraditional.

In the training set, the tweets were classified as:

Label	Count	Percent
Not relevant	242	31.5%
Neutral	257	33.4%
Negative	135	17.4%
Positive	134	17.6%
Total	768	100%

And in the testing set, the tweets were classified as:

Label	Count	Percent
Not relevant	197	31.5%
Neutral	284	33.4%
Negative	152	17.4%
Positive	135	17.6%
Total	768	100%

These are important baseline characteristics for the models to be compared to.

The data that I found departed significantly from what Urban Institute found. In the Urban Institute work, about 2% of tweets were classified as positive; I found a much higher rate. I do not know why this would be the case; to investigate, I would like to look at national-level data in the future in order to assess whether I can reproduce distributions similar to the Urban Institute's. For now, I can hope that this data was skewed positive due to:

- Community involvement increasing

- Positive police responses to high-publicity crimes
- Lori Lightfoot's election, which may herald police reform

4 Modeling

I tried several ways to gain insight on and model the data. In this section of the report, I will first walk through some general observations that I have about the data, then briefly summarize the models that I ran, and then describe future work that I could do to improve these predictions.

I ran a lot of models (on the order of 50, if you include changing hyperparameters) and will report on the overall trends that I found. Additional detail that is less salient can be found in my code.

4.1 General observations about the data

4.1.1 Word counts

I first wanted to get a good sense of what in particular was in my data. First, I looked at words that occurred most commonly across tweets:

Word	Number of tweets included in
chicago	417
police	410
chicagopolic	119
pregnant	96
babi	87
woman	86
say	69
miss	69
found	65
womb	58
synagogu	54
cut	48
cwbchicago	40
dead	40
arrest	38
amp	38
report	37
peopl	35
home	35

Word	Number of tweets included in
charge	32

Some of these words are general, like `chicago` and `chicagopolic` and `arrest`. Others relate to specific news stories, like `pregnant` and `synagogu` (there was a murder of a pregnant woman and a molotov cocktail thrown at a synagogue the week that I pulled tweets). This hints at a larger problem with the data: any week-long time period is not sufficiently general to build a general model on. This was good information for me to learn.

This effect is even more pronounced when bigrams are considered. The most popular bigrams here were: `babi cut`, `chicago police`, `cut womb`, `cwbcchicago chicagopolic`, `found dead`, `miss pregnant`, `polic say`, and `pregnant woman`.

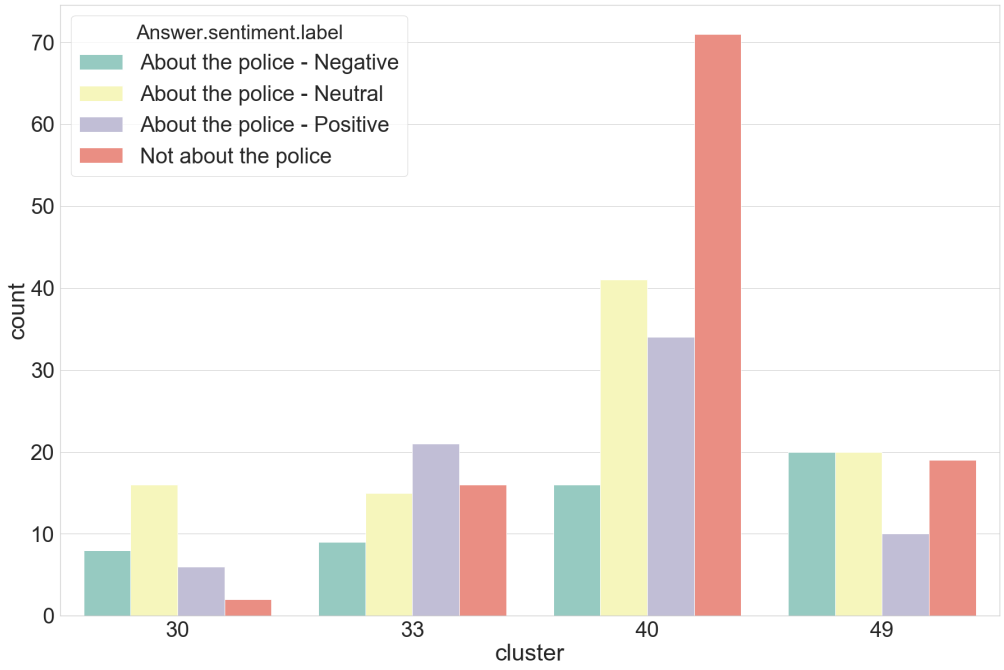
Another important note is that these tweets varied greatly. Even the most common words appeared in only about half of tweets. This led me to cluster my data.

4.1.2 Similarity and clustering

In order to further understand how similar and different tweets are, I next used cosine similarity on a TF-IDF matrix from my data in order to assess how similar my tweets were to each other. The short answer is that most tweets are different from most other tweets.

To investigate this further, I used my similarity matrix to build clusters. This was highly influenced by a tutorial that I saw online, and this tutorial is cited in my code.

More than half of my clusters included fewer than 10 tweets, which was not ideal. The most popular clusters, also, were less than ideal. They showed mixed sentiment by cluster, and the most dominant cluster predominantly had irrelevant tweets. The results from the four most popular clusters are shown below, where the number of tweet per label by cluster is shown.



Clusters with more than 20 members

These results are otherwise difficult to interpret, as many of the processed tweets are difficult to comprehend.

4.2 Results

I considered two types of models, Neural Networks and more traditional models. In this section, I will walk through what I tweaked in each type of model. Then, I will show some of the best results from each. Note that additional work can be found in the code base.

For all models, I looked at three types of labels:

- Four-way classification: similar to how Urban Institute classified their tweets, particularly useful if it can be broadly applied to tweets at large.
- Binary opinion vs. not opinion classification: to reveal which tweets are actually relevant to further analysis.
- Negative sentiment vs. not negative sentiment: to reveal which tweets consider the police negatively.

I chose these labels as I believe they would be most useful to social science researchers. In addition to the Urban Institute work referenced above, other work at Carnegie Mellon has looked at comparing amounts of different types of sentiment in tweets to measure public opinion. In this vein, identifying which tweets are opinion and which are negative opinion could positively impact social science research.

As a brief summary, please find the rough "best" accuracy for each type of model below:

Type	Labels	Neural Net	"Traditional"
4way	positive, negative, neutral, not relevant	~56%	~42%
Binary	opinion, not opinion	~63%	~68%
Binary	negative sentiment, other	~85%	~80%

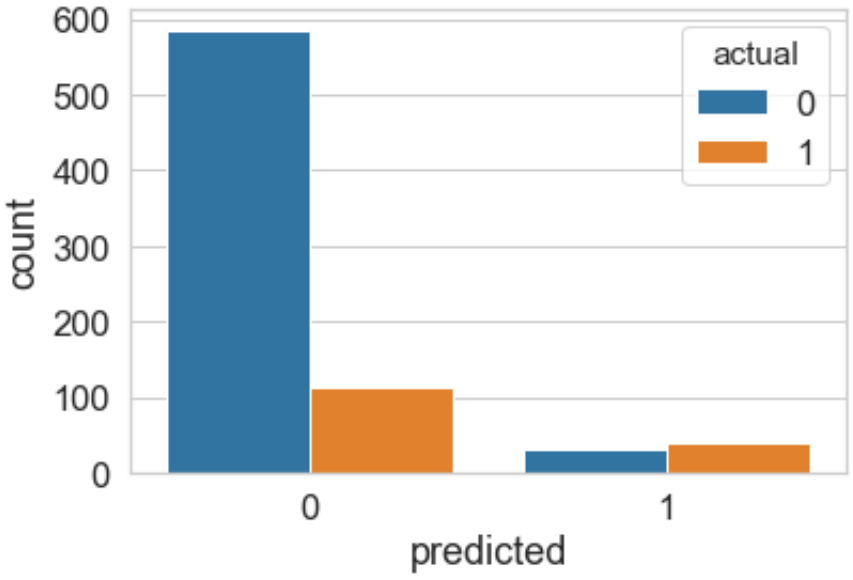
4.2.1 Traditional models

I ran Logistic Regression and Naive Bayes models with the following types of features:

- word bigrams
- word unigrams
- tf-idf bigrams
- tf-idf unigrams

Naive Bayes models out-performed Logistic Regression models; a subset of model results will be shown below.

Classifying a tweet as negative or not negative The Naive Bayes (Multinomial) with unigram features was among the most successful, with a precision of 25.7% and an accuracy of 81.1%. The precision truly does impede the usefulness of the work, however. In the graph below, the number of tweets per predicted labels, split by true labels, is shown. As you can see, the prediction does show improvement over "random guessing", but not by a great deal.



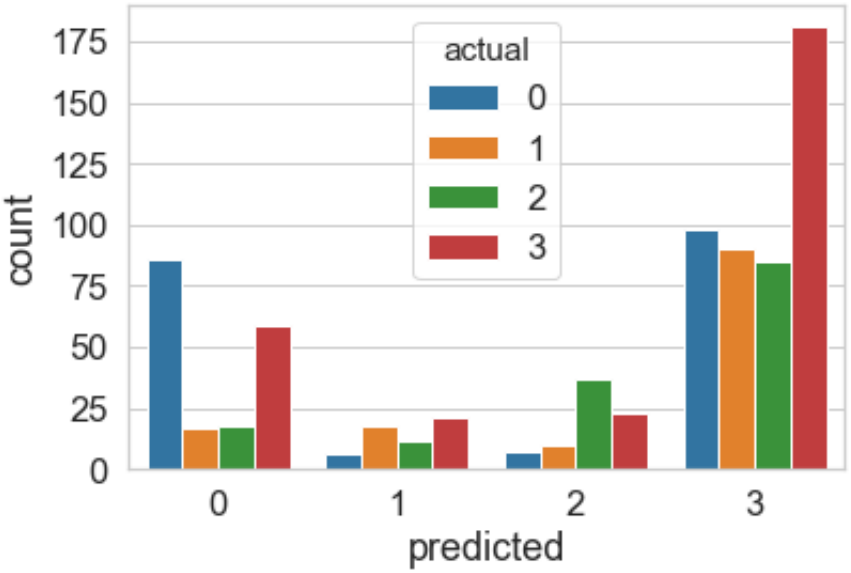
Unigram Naive Bayes Classifier for negative / not negative classification

Classifying a tweet into one of four categories In the graph below, results from a Naive Bayes bigram model (the best of those I tested) is shown. The labels pertain to:

- 0: Not relevant
- 1: Positive
- 2: Negative
- 3: Neutral

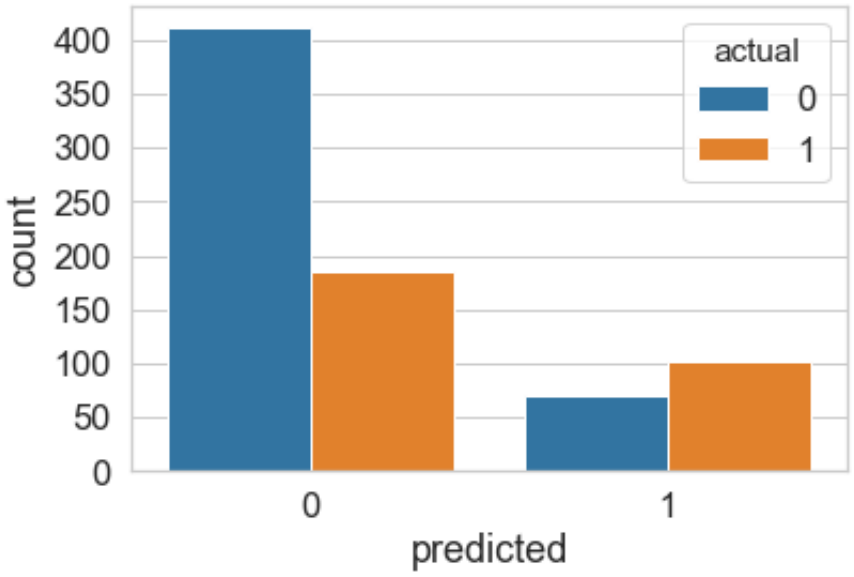
This model shows the most success at predicting neutral tweets. It also performs better than random. In particular, this model had a precision of 36.3% and an accuracy of 41.9%.

I believe that one reason neutral tweets are easier to predict is that they have somewhat similar bigrams. For example, many tweets about news headlines include the term "police said".



Bigram Naive Bayes Classifier for four-way classification

Classifying a tweet as an opinion or not As above, I used an Naive Bayes model based on bigram features to classify a tweet as an opinion or not. This model had precision 35.2% and accuracy 66.7%. The results are shown in a graph below. This model was moderately successful - it was able to predict non-opinion tweets reasonably well, especially given the limited data.

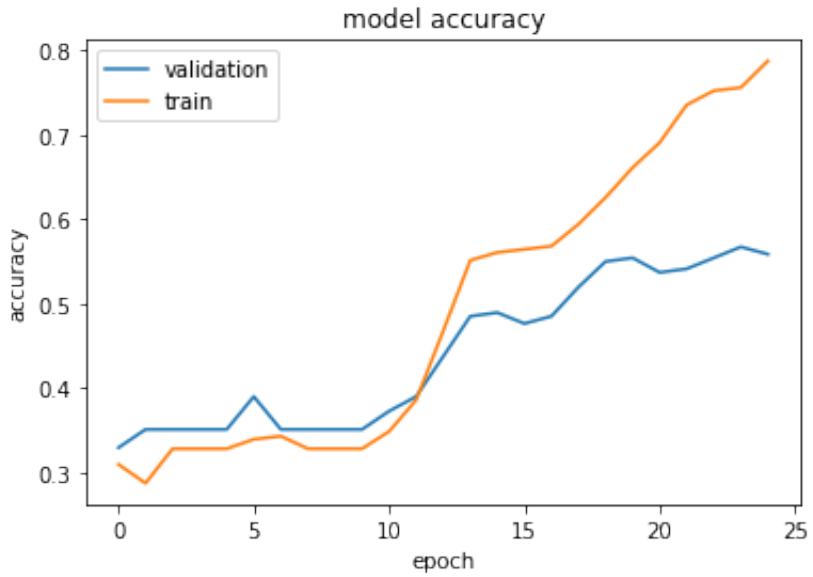


Bigram Naive Bayes Classifier for opinion / not opinion classification

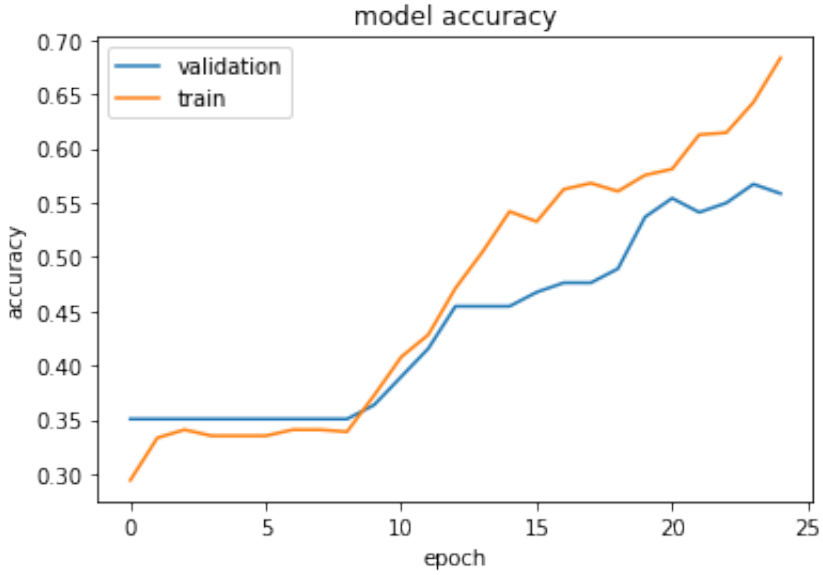
4.2.2 Neural Net models

Neural nets proved to be the wrong tool for this job, due largely to the limited data. These models performed quite poorly; rather, they overfit data extremely. In addition, they were not robust; results vary significantly run-to-run. It seems that much of the models' successes are based on learning the dataset entirely and fitting to it; this suggests the models could be effective given a larger sample size.

A note on overfitting and low sample size In particular, the model actually seems to overfit data regardless of when drop out layers are included. Consider the two figures below. In the first, a four-layer model with no drop out layers is shown. In the second, the same model, but this time, with drop out layers, is shown. The accuracy between validation and test sets does not decrease materially when drop out layers are included. These figures are from neural nets that use word bigrams to predict a 4-way classification.



NN Classifier for 4-way classification without dropout layers

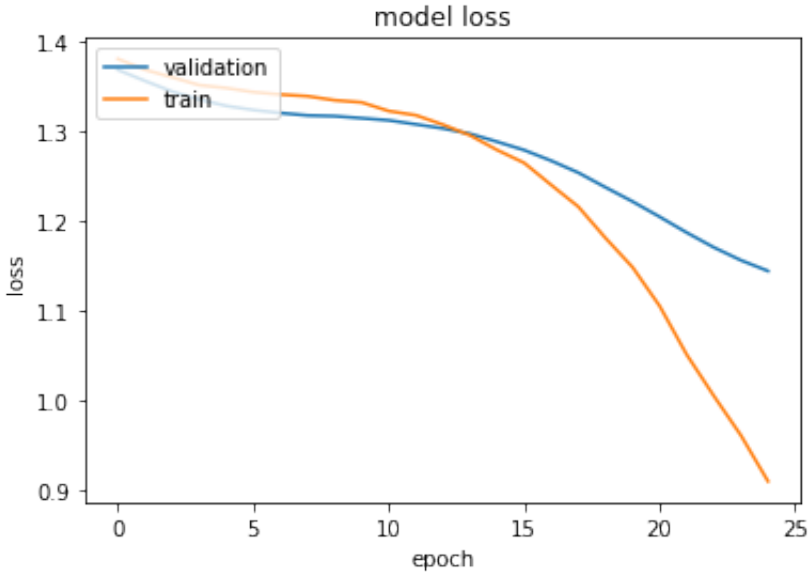


NN Classifier for 4-way classification with dropout layers

It is important to note that this behavior suggests that there simply is not robust enough data to build a neural net well.

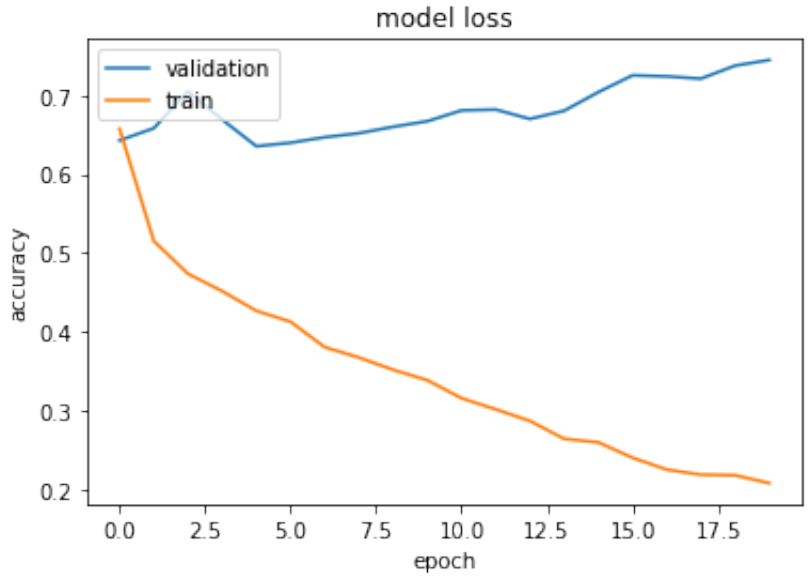
Four way prediction using 4-layer NN classifier The loss over epochs for a neural network with four layers (activations: softmax, relu, softmax; optimizer: adam, loss: categorical cross entropy) are shown below. The accuracy is already shown above. Notice that overfitting is also shown here (loss for validation does not decrease with the training set).

In addition, as shown above, at the end of 25 epochs (ended because overfitting grows over time), there is reasonable accuracy among both training and validation datasets. This is promising relative to traditional models.

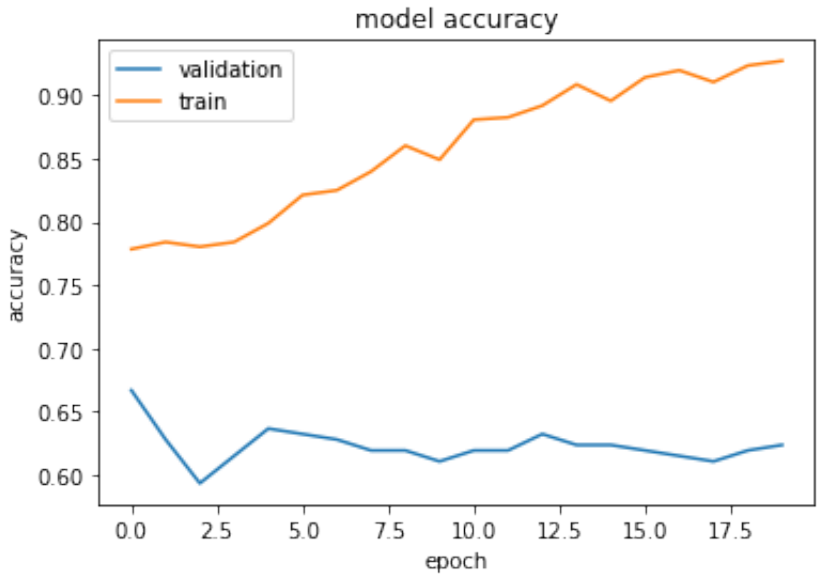


NN Classifier for 4-way classification, loss

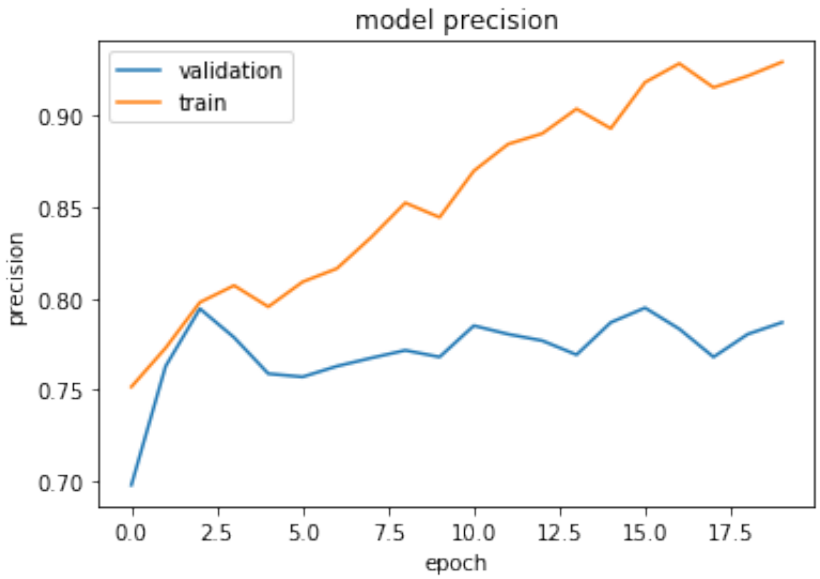
Two-way classification: opinion or not The graphs of precision, accuracy, and loss for a neural network is shown below. This neural network had three dense layers and two drop out layers. It used activation functions softmax, relu, softmax, loss function crossentropy loss, and the adam optimizer. As above, although this overfits training data, it shows promise. With a larger and more varied dataset, a similar neural network could have potential.



NN Classifier for sentiment classification, loss



NN Classifier for sentiment classification, accuracy



NN Classifier for sentiment classification, precision

In particular, notice that the precision far outpaces that of the traditional models. This may simply be an effect of overfitting.

Additional models I also built additional models and played with other parameters for the neural nets, like their optimizer, their loss functions, the number of layers and drop out layers, and the types of activation functions. This dataset, though, was simply too volatile to overwhelmingly conclude what was the best and the worst options. Because with every iteration of this model, results varied significantly, I cannot say which models are the best overall. Instead, I will assert that the models shown above were, to my mind, most promising.

4.3 Discussion and future work

Neural Nets hold the most promise for this work, but overfit data significantly. Though they have significantly better precision over traditional models, there is need for models trained on larger and more varied data.

Future directions for this work include:

- Detecting sarcasm in this corpus (this is a deep area of research that I was unable to fully delve into in such a short period of time)
- Using data from more weeks and more cities
- Using more data in general

5 What did I learn for this project?

To complete this project, I had to familiarize myself with a number of technologies, among other material (like learning more about types of models):

1. keras
2. MTurk
3. using sklearn for text analysis
4. the twitter API

In particular, using MTurk took a ton of time!

6 Selected sources

6.1 Papers

- Ekins, Emily. (2016). Policing in America: Understanding Public Attitudes Toward the Police. Results from a National Survey. SSRN Electronic Journal. 10.2139/ssrn.2919449.
- Fowler, AF Rengifo and K. 2016. "Stop, Question, and Complain: Citizen Grievances Against the NYPD and the Opacity of Police Stops Across New York City Precincts, 2007-2013." Journal of Urban Health (93 Suppl 1): 32-41.
- O'Connor, Brendan & Balasubramanyan, Ramnath & R. Routledge, Bryan & A. Smith, Noah. (2010). From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. International AAAI Conference on Weblogs and Social Media. 11.
- Skogan, Wesley G. 2006. "Asymmetry in the Impact of Encounters With Police." Policing and Society.
- Tyler, Tom R. 2004. "Enhancing Police Legitimacy." The Annals of the American Academy of Political and Social Science 593: 84-99.

6.2 News and articles (quick links)

- https://www.washingtonpost.com/graphics/2018/investigations/unsolved-homicide-database/?utm_term=.8da8a801878a&city=indianapolis]
- https://chicago.suntimes.com/news/murder-clearance-rate-in-chicago-hit-new-low-in-2017/
- https://www.theatlantic.com/ideas/archive/2018/05/quis-custodiet-ipsos-custodes/560324/
- https://datasmart.ash.harvard.edu/news/article/map-monday-unsolved-homicides