Checkpoint 5: Natural Language Processing

The Silent Foxes

Our theme is to explore the relationship between the misconducts in police officers and their career development.

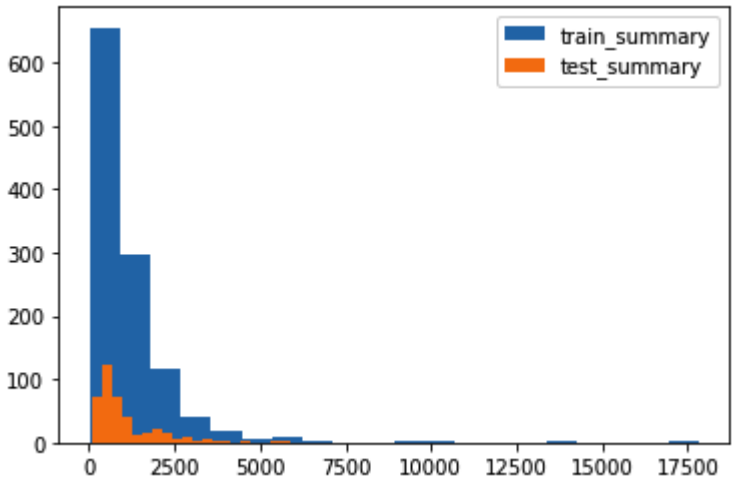
Questions List:

1. Can we train a transformer-based language model to identify the officer complaint from allegation narratives?
2. Is the number of officer complaint records in allegation narratives consistent with existing data?

Q1: Can we train a transformer-based language model to identify the officer complaint from allegation narratives?

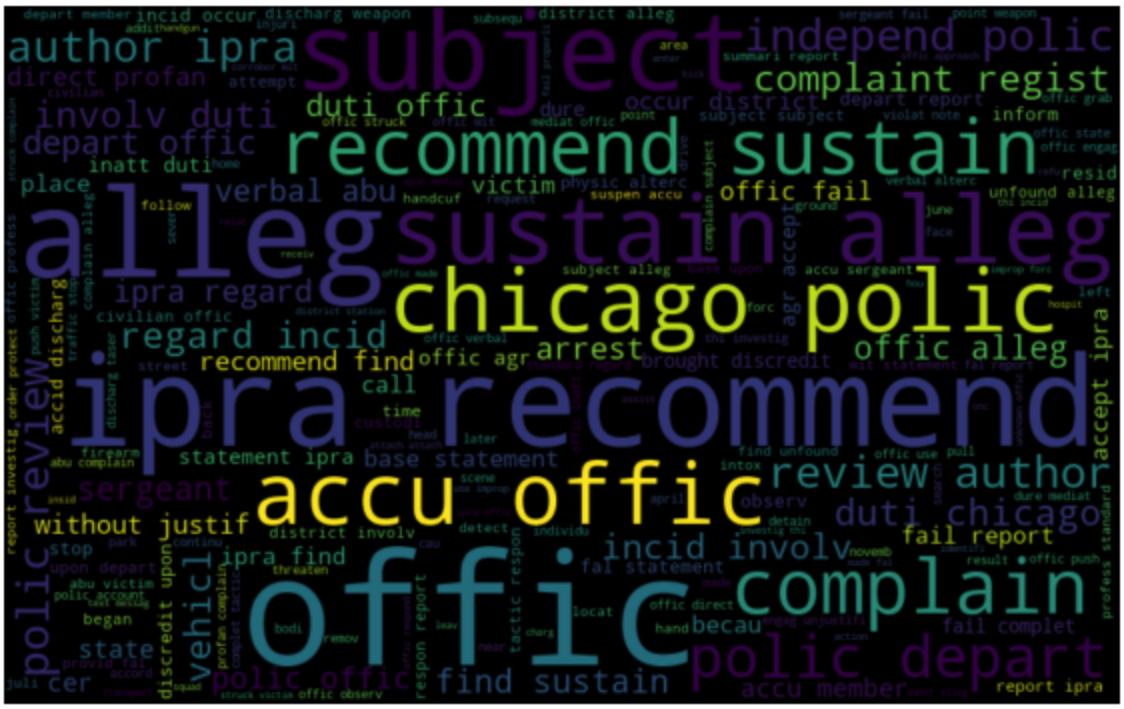
In the previous analysis, we used a field called if_officer_complaint a lot in the table data_allegation, which is an identifier if an allegation is about an officer complaint. Here we checked the table data_allegation especially summary and if is_officer_complaint. Therefore, we think it is helpful for later data exploration to build a model that can automatically fill out this field from allegation narratives.
First, for choosing the dataset for training and testing, we extract 'summary' and 'is_officer_complaint' fields from table data_allegation, where the summary is not empty.
Our dataset is quite small, there are only 1147 pieces of data containing actual summary in table data_allegation. For these data, 772 allegations are not officer complaints and 375 are officer complaints. We take the first 70% data as training

data and the last 30% as test data. The following graph shows a comparison of their summary data length.



The next step is text cleaning. Text contains several kinds of noises, which are unnecessary in training and may mislead the machine to understand the patterns in the input text. First, we normalize the text by removing special characters, like hashtags, numbers as well as short words. Stopwords, such as I, the, often use frequently in a sentence, but provide little knowledge, which is necessary to remove them. Stemming and lemmatization is used to reduce words to their root form. We use some models, like nltk and wordninja, to implement data cleaning.

Next we can use the word cloud to understand the common words used in the summary.
For all summary, we get its word cloud as this:



For summary with officer complaint, we get its word cloud as this:

For summary without officer complaint, we get its word cloud as this:

From these word clouds, we may assume the word “complaint” may be a key aspect that the machine uses to identify patterns.

Then, we extract features from the cleaned text. We have two features: bag-of-words and TF-IDF. The bag-of-words works in a way that we regard each summary as a bag, and in each bag, we count the number of appearances for each work. The TF-IDF features are calculated by term frequency and inverse document frequency, how common or infrequent a word shows in the entire text.

```
(0, 764)      3
(0, 52)       1
(0, 268)      1
(0, 262)      1
(0, 724)      6
(0, 723)      1
(0, 527)      3
(0, 167)      6
(0, 755)      9
(0, 387)      9
(0, 96)       1
(0, 837)      7
```

```
(0, 729)      0.0456
(0, 542)      0.0505
(0, 501)      0.0408
(0, 917)      0.0629
(0, 983)      0.0699
(0, 820)      0.0518
(0, 65)       0.0435
(0, 717)      0.0402
(0, 196)      0.0402
(0, 419)      0.0501
(0, 457)      0.0397
(0, 892)      0.0335
```

The left one is the structure of bag-of-words features, and the right one is TF-IDF features. (0, 764) means no. 0 bag of words, word vector 764 appears three times; same as frequency on right.

Then we implement the gensim Word2Vec model and see how it performs. We will specify a word and the model will pull out the most similar words from the corpus. Details can be checked in the Jupyter notebook.
Finally, we tried logistic regression. We use Bag-of-Words features and split data into training and validation sets. We use F1 scores to represent the accuracy of our model, which is a common measurement for precision and recall of the test.

Feature	Bag-of-Words	TF-IDF
Accuracy	0.6956521	0.6945606

Conclusion: There is no big difference between using two features, which shows the accuracy of the model is around 70%. Hence, our training model can predict if it is an officer complaint according to allegation narratives in an above average accuracy.

Q2: Is the number of officer complaint records in allegation narratives consistent with existing data?

This question actually means that if the percentage of officer complaints in total allegations in existing table data_allegation is similar to our prediction percentage from allegation narratives provided.
First, we calculate the percentage of officer complaints in total allegations. There are two cases. In the table data_allegation, there are only 1147 out of 216052 data that have a summary field(which is not Null). This is what we use for the training model. Also, we calculate the percentage for the entire data_allegation, no matter if it has an empty summary field.
For prediction, we use the provided allegation narratives, in which we extract the text field, and perform text cleaning and apply TF-IDF features. Then, we predict the number of officer complaints in these allegations using our trained model. After

data cleaning, we have 21303 allegation reports, and we get there are 4016 of them which are officer complaints.

Dataset	Total allegations	Is_officer complaint = True	Percentage
With summary	1147	375	0.32693
With/without summary	202608	13444	0.06635
Prediction	21303	4016	0.18851

We can see that 18 percent of allegations are officer complaints predicted from narratives. This percentage is not perfectly consistent with existing data, but we think this makes sense. Because the amount of our training data is too small, and the entire dataset is too large compared to training and prediction data. This could influence the accuracy of our model. Moreover, our prediction lies between the small dataset and the large dataset, so we think the model works properly. If we can have more data to feed the model in the training process, the accuracy of the model would be higher.