Luke Salamone

Simon Benigeri

Renpin Luo

Checkpoint 5: Natural Language Processing

For our NLP checkpoint we looked at the following question:

> For complaint report narratives, is the text predictive of the complaint category?

Prediction of category for complaints reports narratives, looking at Allegation and Initial / Intake Allegation separately

Script: checkpoint_5_data.sql (generates the crid map csv)
Data: narratives csv url, crid map csv url
Notebook: checkpoint_5_fasttext.ipynb

Context

Our group has been investigating the relationship between officer allegations and rewards within the Chicago Police Department, and has previously found troubling evidence that the two appear to be positively correlated. For checkpoint 1, we showed that in many years, officers with up to 5 allegations outperformed officers with no allegations in a given year. In checkpoint 2, we expanded upon this finding by illustrating that units with the most awards were the most frequently decorated. In checkpoint 3 our group illustrated a similarly troubling relationship between pay and sustained allegations, that as the number of allegations increases, the mean pay also increases. In checkpoint 4, we predicted officer pay raises.

For checkpoint 5, we struggled to come up with an NLP task that fit with our theme. Given the narratives dataset, we decided to look into the categorization of complaint report narratives. In seeing if we can predict the category of a complaint report narrative given the narrative's text, we hope to gain some insight into the quality of the categorization.

The data

1. Labels for the 2 Datasets: Complaint report ID to Category Mapping

We queried the CPD database for crids and their respective categories. From the resulting CSV, we get a mapping of 1 category to n crids.

2. Dataset 1: intakes

We read the narratives csv into a dataframe and created a data set from the texts of narratives with column_name = intakes / initial allegations. For each complaint report id, there may be more than one text sample. We found that the text samples were very similar. The difference was often down to a typo, a few extra characters, or small differences in the passage. So from this CSV we constructed an intakes dataset, by selecting the first text passage for each complaint report id, labelled with the corresponding category. We also clean up the text to some degree.

3. Dataset 2: allegations

We do the same thing as with Dataset 1, but with column_name = allegations.

Methodology

Using the fastText library.

We use the Fasttext library in this checkpoint. So in the notebook, we manipulate the data so that it is in csv form where each row is : “__label__n”, “text sample”, as required by fastText. Where n is the integer representation of the class, but cast as a string.

We split our Training and Test data with an 80/20 ratio. We call these splits dataset_TRAIN and dataset_test. We then split dataset_TRAIN into dataset_train and dataset_val, also with an 80/20 ratio. For both datasets we optimize the hyperparameters of a Fasttext supervised model by training on dataset_train and measuring performance on dataset_validation. For the final model and reporting of results, we use the optimal hyperparameters, train on the full training dataset, dataset_TRAIN and test on dataset_test, a blind test dataset.

Why did we use fastText?

fastText is a library that can get us quickly up and running with our own word vector representation of the text in narratives. And we can use this representation for text classification. fastText can create representations of words that do not exist. Given the poor spelling often encountered in narratives, the occurrence of uncommon, domain specific terms, as well as errors due to OCR text extraction, we felt that this functionality would be useful.

fastText supervised training has an auto tune functionality. So given dataset_train and dataset_val, we were able to find good hyperparameters with relative ease.

We hoped to use a pre-trained fasttext english model as a base, and to fine tune it on our narratives data. Unfortunately we struggled with this task. We ran out of RAM in google colab. A solution would have been to download a model, reduce the number of dimensions, save that somewhere where we can pick it up and use it as input to the fastText supervised learning function, but we did not have time to go down that path.

Analysis

Dataset: Intakes

	train_precision	train_recall	train_f1-score	train_support	test_precision	test_recall	test_f1-score	test_support
Bribery / Official Corruption	0.000000	0.000000	0.000000	25	0.000000	0.000000	0.000000	2
Conduct Unbecoming (Off-Duty)	0.693878	0.490385	0.574648	208	0.533333	0.266667	0.355556	60
Criminal Misconduct	1.000000	0.129032	0.228571	31	0.000000	0.000000	0.000000	9
Domestic	0.000000	0.000000	0.000000	65	0.000000	0.000000	0.000000	17
Drug / Alcohol Abuse	1.000000	0.066667	0.125000	30	0.000000	0.000000	0.000000	7
False Arrest	0.799472	0.767089	0.782946	395	0.774194	0.642857	0.702439	112
First Amendment	0.000000	0.000000	0.000000	1	0.000000	0.000000	0.000000	1
Illegal Search	0.802071	0.864764	0.832239	806	0.689956	0.763285	0.724771	207
Lockup Procedures	0.821530	0.810056	0.815752	358	0.534884	0.494624	0.513966	93
Operation/Personnel Violations	0.805247	0.964324	0.877635	1878	0.686515	0.871111	0.767875	450
Supervisory Responsibilities	0.000000	0.000000	0.000000	20	0.000000	0.000000	0.000000	4
Traffic	0.769231	0.058140	0.108108	172	0.500000	0.027027	0.051282	37
Use Of Force	0.620690	0.666667	0.642857	243	0.559322	0.550000	0.554622	60
Verbal Abuse	0.000000	0.000000	0.000000	45	0.000000	0.000000	0.000000	11
accuracy	0.790507	0.790507	0.790507	0	0.671028	0.671028	0.671028	0
macro avg	0.522294	0.344080	0.356268	4277	0.305586	0.258255	0.262179	1070
weighted avg	0.761535	0.790507	0.754140	4277	0.628286	0.671028	0.634160	1070

Dataset: Allegations

	train_precision	train_recall	train_f1-score	train_support	test_precision	test_recall	test_f1-score	test_support
Bribery / Official Corruption	0.000000	0.000000	0.000000	15	0.000000	0.000000	0.000000	5.0
Conduct Unbecoming (Off-Duty)	0.238095	0.170940	0.199005	117	0.560000	0.500000	0.528302	28.0
Criminal Misconduct	0.000000	0.000000	0.000000	31	0.000000	0.000000	0.000000	5.0
Domestic	0.702703	0.419355	0.525253	62	0.166667	0.083333	0.111111	12.0
Drug / Alcohol Abuse	1.000000	0.120000	0.214286	25	1.000000	0.100000	0.181818	10.0
False Arrest	0.823171	0.912162	0.865385	148	0.673913	0.861111	0.756098	36.0
First Amendment	0.000000	0.000000	0.000000	1	NaN	NaN	NaN	NaN
Illegal Search	0.720497	0.852941	0.781145	136	0.472222	0.531250	0.500000	32.0
Lockup Procedures	0.000000	0.000000	0.000000	46	0.000000	0.000000	0.000000	10.0
Operation/Personnel Violations	0.664384	0.944805	0.780161	308	0.608000	0.853933	0.710280	89.0
Supervisory Responsibilities	0.000000	0.000000	0.000000	4	0.000000	0.000000	0.000000	4.0
Traffic	0.000000	0.000000	0.000000	23	0.000000	0.000000	0.000000	4.0
Use Of Force	0.832061	0.956140	0.889796	228	0.727273	0.769231	0.747664	52.0
Verbal Abuse	0.714286	0.609756	0.657895	41	0.666667	0.200000	0.307692	10.0
accuracy	0.703797	0.703797	0.703797	0	0.612795	0.612795	0.612795	0.0
macro avg	0.406800	0.356150	0.350923	1185	0.374980	0.299912	0.295613	297.0
weighted avg	0.624361	0.703797	0.646123	1185	0.557740	0.612795	0.560047	297.0

The problem is mainly class imbalance. For both datasets, we seem to overfit on overrepresented classes because the classifiers achieve high precision, recall and f1 on the training set, and the test set. Predictably, we underfit on the underrepresented classes. For these classes, we have very few samples, therefore we don't have a good enough

representation of the variety that can exist in them. Also, the severe overrepresentation of certain classes skews results in their favor.

Given this experiment, it's hard to say much about the quality of narratives categories. Our models suffered from class imbalance.

Future Research

As far as this particular checkpoint goes, we can improvements in a number of ways:

1. We could use a more balanced dataset, either by restricting the number of classes to the well represented classes, or by gathering more data for the underrepresented classes.
2. We could find a more intelligent way to handle underrepresented classes. For example, by using rare even specification techniques, such as anomaly detection. We are aware that with credit card fraud detection, the significant class, “FRAUD”, is underrepresented. It is probable that we find useful techniques and ideas from these types of problems.
3. We could fine tune a pretrained fasttext model using this dataset. Earlier we describe issues we had with implementing this with fastText in google colab. There are many fine tuning resources which we could have used here.

On a less technical level, we observed that the category of a complaint record says more about the circumstances than the act. For example, an officer verbally abuses an individual at a traffic stop. The category was “Traffic”. If the narrative describes the act, but the category describes the circumstances, then is the language used in the narrative going to be predictive of the category? We saw another group use topic modeling as a solution to redefine the complaint report categories. Perhaps this is a more useful avenue for future research.