Data Science Seminar: Checkpoint 5 Findings

The Earnest Pirates
Vinit Todai, Shreyas Lele, Tejul Pandit
Checkpoint 5: Natural Language Processing

NLP Question -

Amongst officers who belong to a rank that is in the top 10 ranks with the maximum allegation counts, what are the different topics that can be modelled on the allegation text?

- We filtered the officer data based on the ranks and count of allegations; the corresponding allegation text against the officers identified is used to assess the different topics of allegations covered.

Results -

Amongst officers who belong to a rank that is in the top 10 ranks with the maximum allegation counts, what are the different topics that can be modelled on the allegation text?

Topic modelling is a statistical technique that is used to identify and discover distinct "topics" present in a series of documents or texts. The goal for Checkpoint 5 was to identify the different clusters of topics covered from the allegations that are registered against a police officer. The vision was further narrowed down to target only those police officers who had maximum number of complaints against them.

The text consisting of allegations against the police officers is present in the cr_text column in *data_allegation* table. For the different police officers identified with maximum count of allegations multiple allegation ids are mapped which is further used to extract the corresponding allegation text.

We begin the code by first installing and importing required packages followed by connecting to the database and fetching the necessary data. The dataset is further split into train and test subsets wherein the train set is used for developing the model for topic clustering and the last 10 records are taken for testing the model created.

Later, the following steps are covered in the code -

1) Pre-process and clean the text.
2) Create the topic model.
3) Visualize the model created.
4) Change the number of clusters to be made while topic modelling and re-train the model.
5) Visualize the new model created.
6) Save the model.
7) Test the topic model on unseen data to assign groups to new text.

Section 1: Pre-process and clean the text

A sample raw text present in the table is presented in Figure 1. The text consists of the different allegation intakes which have additional information added with successive iterations to form the final text. As observed in Figure 1, special characters such as '\n', punctuations, and digits are present which is redundant for topic clustering.

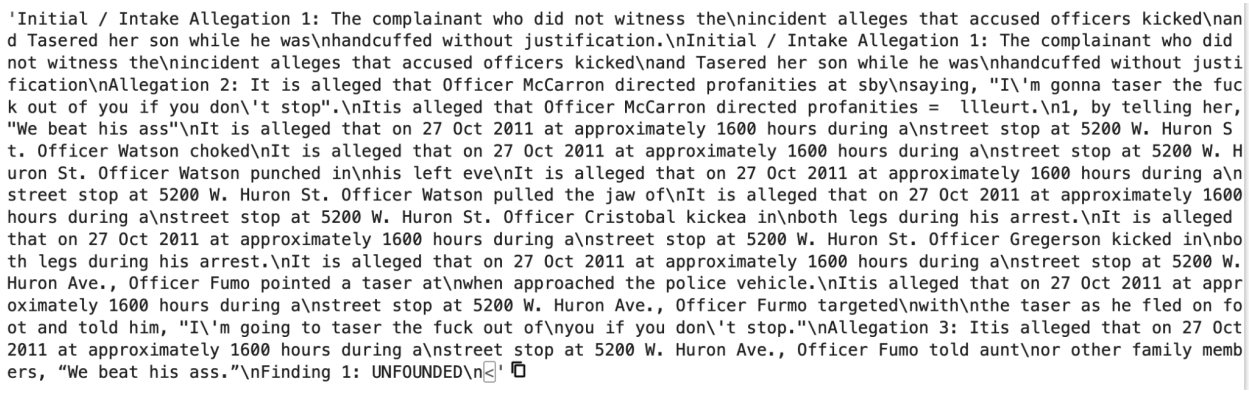

Figure 1: Raw sample text from cr_text column in Database.

The preprocessing on raw text is covered in a single function. The text is first split by the ':' mark and the first statement is selected which consists of the actual complaint. Rest information preceding and post this statement is dropped since most of the information regarding the allegation is included in the first statement and post that details to the complaint are added. Additionally, regex is used to remove characters and phrases like '\n' , 'Initial / Intake Allegation', digits, and punctuations. NLTK is a powerful Python Natural Language library that consists of various functionalities of which we utilize stopwords and PorterStemmer. Using stopwords, words such as - i, me, my, also, a, the, an, and, so, etc. are removed from the corpus. Furthermore, PorterStemmer is a stemming technique. This helps to identify the root word of each word due to which different variations of words such as - justification and justified - are brought to its root form - justif.

The pre-processing function is applied on all the rows of allegation_text. The resultant text for the sample is Figure 1 is shown in Figure 2.

```
'the complain wit incid alleg accus offic kick taser son handcuf without justif'
```

Figure 2: Pre-processed text of raw sample text shown in Figure 1

Section 2: Create the topic model

Topic modeling is an unsupervised machine learning technique that automatically identifies different topics present in a document (textual data). For our code, we use the Bertopic algorithm. BerTopic is a topic modeling technique that uses transformers (BERT embeddings)

and class-based TF-IDF to create dense clusters. The following sequence of steps are performed when bertopic() is fit and transformed on our cleaned data -

1) Create sentence embeddings using BERT.
2) Cluster sentences into semantically similar clusters.
3) Create topic representations from clusters.

Without defining the number of clusters, the model creates 201 groups that signify 209 different topics. A count of samples in each cluster is presented in Figure 3. Also, the 'Name' column presents the topic number followed by top-4 most occurring words in the corresponding cluster. Topic -1 refers to the set of sentences that could not be assigned any class and are outliers for the data at hand.

```
topic_model.get_topic_info()
```

	Topic	Count	Name
0	-1	1325	-1_falsifi_retaili_chicago_broke
1	0	82	0_rude_disrespect_speak_shirt
2	1	62	1_smoke_cigarett_harm_cta
3	2	62	2_accid_ticket_traffic_tegard
4	3	60	3_usc_overag_discrep_cashier
...	...	...	...
204	203	11	203_ipra_burglar_password_videotap
205	204	11	204_did_wit_minor_twice
206	205	11	205_abduct_mother_daughter_dog
207	206	11	206_repli_stroller_homeless_restaur
208	207	11	207_cousin_devic_explan_order

209 rows × 3 columns

Figure 3: Information on different topics created

Figure 4 provides further information regarding a single cluster, 4, wherein the top words and their corresponding class-based TF-IDF scores are populated. On observing Figure 4, the topic covered by this cluster consists of words such as landlord and evict which directs towards complaints regarding landlord and tenant issues.

```
topic_model.get_topic(4)
```

```
[('landlord', 0.059059504118049444),
 ('tenant', 0.03651305871908606),
 ('grow', 0.01875900169555575),
 ('basement', 0.018139521480176345),
 ('rent', 0.013702569929593355),
 ('build', 0.012414010401981115),
 ('evelyn', 0.010118972508132713),
 ('evict', 0.009241354960761106),
 ('henderson', 0.0072128384459608594),
 ('landlordten', 0.0072128384459608594)]
```

Figure 4: Analysis of topic 4

Section 3: Visualize the model created

The topics generated can be visualized by using the visualize_topics() function on our model. This can be seen in Figure 5. Various clusters are created and their sizes differ based on the count of instances present in each cluster. Also, on hovering over the image, top words are also displayed. Text corresponding to topic -1 is ignored while creating the visualization.

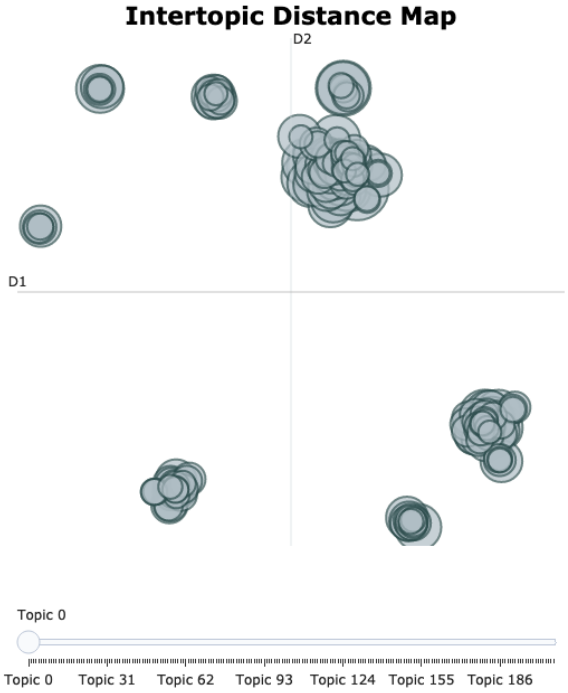

Figure 5: Distance map of the different (209) topics created

Section 4: Reduce number of clusters

On observing Figure 5, it can be seen that multiple clusters overlap each other. As a result, similar topics can be grouped together to reduce the number of clusters. For the updated model, nr_topics i.e. number of topics are restricted to 6 in the Bertopic() function which is then fit and transformed onto the allegations text. The updated cluster information is shown in Figure 6.

```
topic_model.get_topic_info()
```

	Topic	Count	Name
0	-1	1419	-1_report_parti_victim_polic
1	0	1248	0_alleg_accus_victim_complain
2	1	1111	1_report_parti_accus_state
3	2	859	2_offic_report_door_parti
4	3	587	3_report_parti_sergeant_arrest
5	4	468	4_vehicl_search_arrest_victim
6	5	465	5_parti_alleg_inventori_plant

Figure 6: Information on updated topics created

By updating the model, topic 4 now corresponds to a different set of allegations. Figure 7 provides the words that make up the cluster followed by class-based TF-IDF scores. The target words vehicle and search signify the complaints are regarding illegal vehicle search.

```
topic_model.get_topic(4)
```

```
[('vehicl', 0.06328782794183578),
 ('search', 0.05911232458968046),
 ('arrest', 0.05799489396934193),
 ('victim', 0.04596237899730433),
 ('warrant', 0.02818013678229811),
 ('coerc', 0.028153415677698215),
 ('impound', 0.023704699922724427),
 ('permiss', 0.02341500840813096),
 ('detain', 0.023018069050091985),
 ('plaintiff', 0.019480062816899735)]
```

Figure 7: Analysis of topic 4 of updated model

Figure 8 provides a pre-processed sample of text that belongs to cluster 4 to provide further insight into the type of allegations belonging to cluster 4.

```
'the complain alleg accus offic search vehicl without justif the complain alleg accus offic detain without justif find'
```

Figure 8: Sample pre-processed text from cluster 4

Section 5: Visualize the new model created

The updated clusters are visualized to analyze the inter-cluster distances. The map is shown in Figure 9. Due to reduction in the number of topics, no cluster/ topic overlap is present.

Figure 9: Distance map of the different (6) topics created

To gain further insights into each of the topics generated, a barchart consisting of top words that belong in each topic and their corresponding class-based TF-IDF scores are visualized. Figure 10 provides the barchart visualization that can be used to compare the different topics. As mentioned earlier, topic -1 is dropped during visualization as it consists of a group of texts that could not be assigned any class.

Figure 10: Topic-wise bar chart visualization

On observing the bar chart in Figure 10, some topics seem similar despite the increase in inter-topic distances as seen in Figure 9. To visualize how similar topics are to each other, Figure 11 provides a heat map. From Figure 11, darker colored blocks show higher similarity and thus the bright blue colour signifies that topics 0 and 1 and topics 2 and 3 are similar to each other.

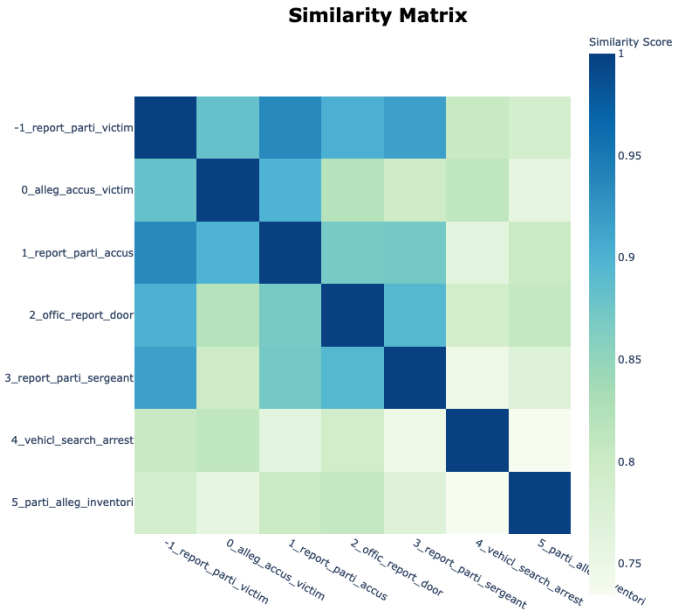

Figure 11: Similarity matrix between different topics

Section 6: Save the model

The final model created is saved which can be later loaded directly into the Python code to test on new data. The model is present in Checkpoint 5/topics_model.

Section 7: Test the topic model on unseen data

The model created is now tested on unseen data to analyze the classes that are assigned to each record in unseen/ test data. Figure 12 provides the final result on 10 records that were not observed by the model before. Out of the 10 observations, 3 observations could not be classified in a specific topic, however, remaining records are classified into respective clusters.

	id	rank	allegation_id	allegation_text	final_topic
0	32424	Police Officer	1056276	Initial / Intake Allegation 1: The reporting p...	1
1	32430	Police Officer	1055255	Initial / Intake Allegation 1: The complainant...	-1
2	32430	Police Officer	1058438	Initial / Intake Allegation 1: THE REPORTING ...	0
3	32430	Police Officer	1058984	Initial / Intake Allegation 1: a warrant and\n...	2
4	32432	Sergeant of Police	1075034	Initial / Intake Allegation 1: THE REPORTING P...	3
5	32433	Police Officer Assigned Evidence Technician	1060383	Initial / Intake Allegation 3: The reporting p...	5
6	32435	Police Officer	1049816	Initial / Intake Allegation 1: The reporting p...	-1
7	32438	Police Officer	1057765	Initial / Intake Allegation 1: The reporting p...	2
8	32442	Police Officer	1057569	Initial / Intake Allegation 1: The reporting p...	0
9	32465	Police Officer	1052346	Initial / Intake Allegation 1: THE REPORTING P...	1

Figure 12: Test data topic classification

Conclusion -

Using the NLP analysis on allegation text, various groups of allegations are tried to be identified. This helps to analyze the kind of allegations that are raised against the police officers with the maximum number of allegations. When the model was updated to have only 6 topic classes, despite having 1419 records that could not be placed into any of the classes, majority records are assigned a topic that signify the type or the class of allegations that are present. Furthermore, as observed in Figure 7, topic 4 aimed at identifying complaints that were related to illegal search of vehicle or of the complainant when the complainant was in a vehicle. Additionally, by creating topic models, it provides further insight in the allegation category assigned to each allegation.

Also, usually it is difficult to determine what could be an individual's next action. With the help of the topic model, a future scope of the project can be used to visualize if any trends are present in the behavior of a police officer based on the allegations received against him/ her to predict and prevent their next action.