Document Classification

Lu Luo Jan 2021
https://github.com/ninjalu/doc-classification

Data and challenges

Data and challenges

Data and objective

- Adarga provided us with 4 series of labelled online articles, a mixture of url links and texts.
- After scraping from the url links and consolidated all the texts, we have: 100 profile reports; 99 situation reports; 88 incident reports and 74 analytical reports.
- The objective is to classify given documents into 4 categories: analytical, situation, profile and incident reports.

Data and challenges

Data and objective

- Adarga told us the reason for classification is that it could help downstream tasks such as named entity recognition.
- We need to bear that in mind as we tackle this.

Data and challenges

Challenges

- The training data comes in different formats. We need to conduct some scraping and preprocessing to make sure data is in the right format for the analysis.
- The training data is relatively small: 361 samples in total.
- Document lengths are very long. For neural NLP this could be a challenge.

Ideas

Ideas

Traditional NLP

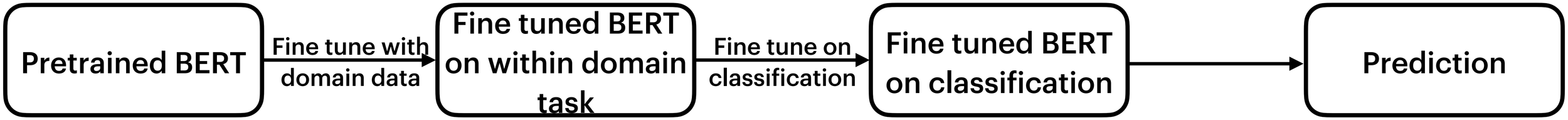
- From the project proposal, we learned some insights of how the documents were classified. This provides guidance to some aspects of the hand-crafted featuring engineering.
- TF-IDF combing with SVD could perhaps provide some useful features of how articles are written.

Situation report	Short syntactical repetition	Large entities; few repeats				
Incident report	Descriptive less repetitive	Fewer entities;				
Profile report	Descriptive	Single entities;	More time	More places		
Analytical report	Cautious				More stats	More cautious words

Ideas

BERT: one

- Fine tune a pretrained BERT model with an additional classification layer.
- Within-domain fine tune (existing corpus, collecting more data?) pretrained BERT model, fine tune again with classification task.



Ideas

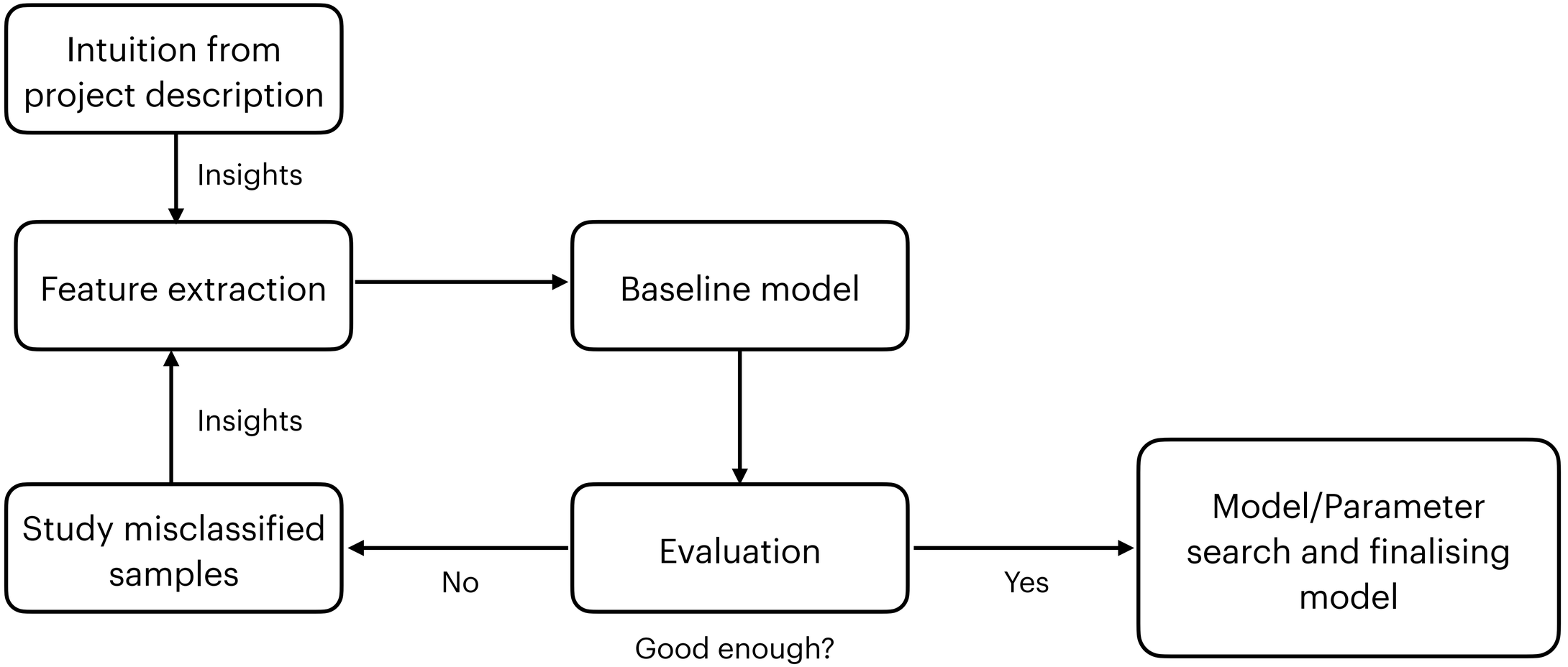
Pros and cons

	PROS	CONS
Traditional NLP	1. Fast to train and to inference 2. Good interpretability 3. Less concerned about document size	1. Do not ‘understand’ language as well as BERT. 2. Feature engineering is based on domain understanding which can be ineffective
BERT	1. Good all around ‘understanding’ of the language. 2. Not relying on domain expertise to produce good results. 3. Pretrained models can benefit downstream tasks	1. Slow to train and inference. 2. Document size could be a concern 3. Black box difficult to interpret.

Traditional NLP

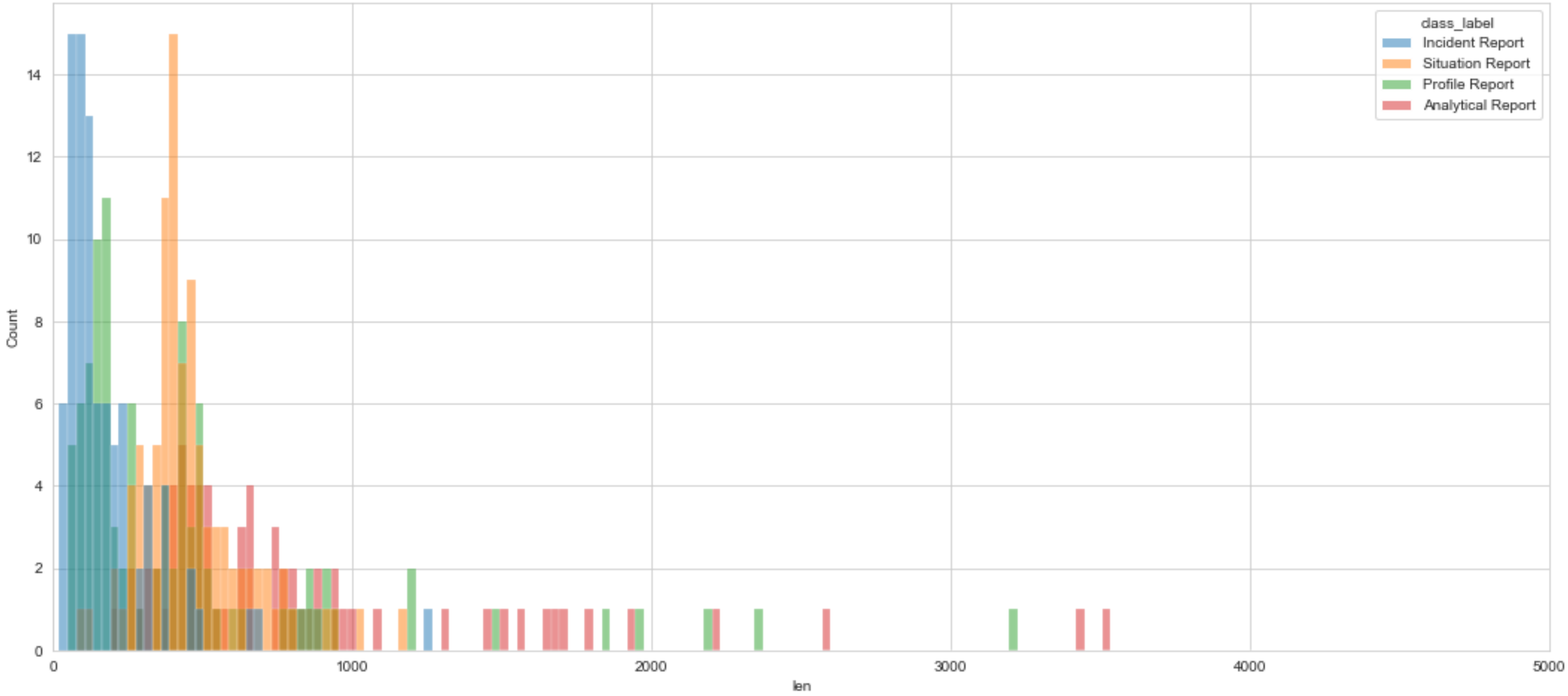
Traditional NLP

My workflow for handcrafting features



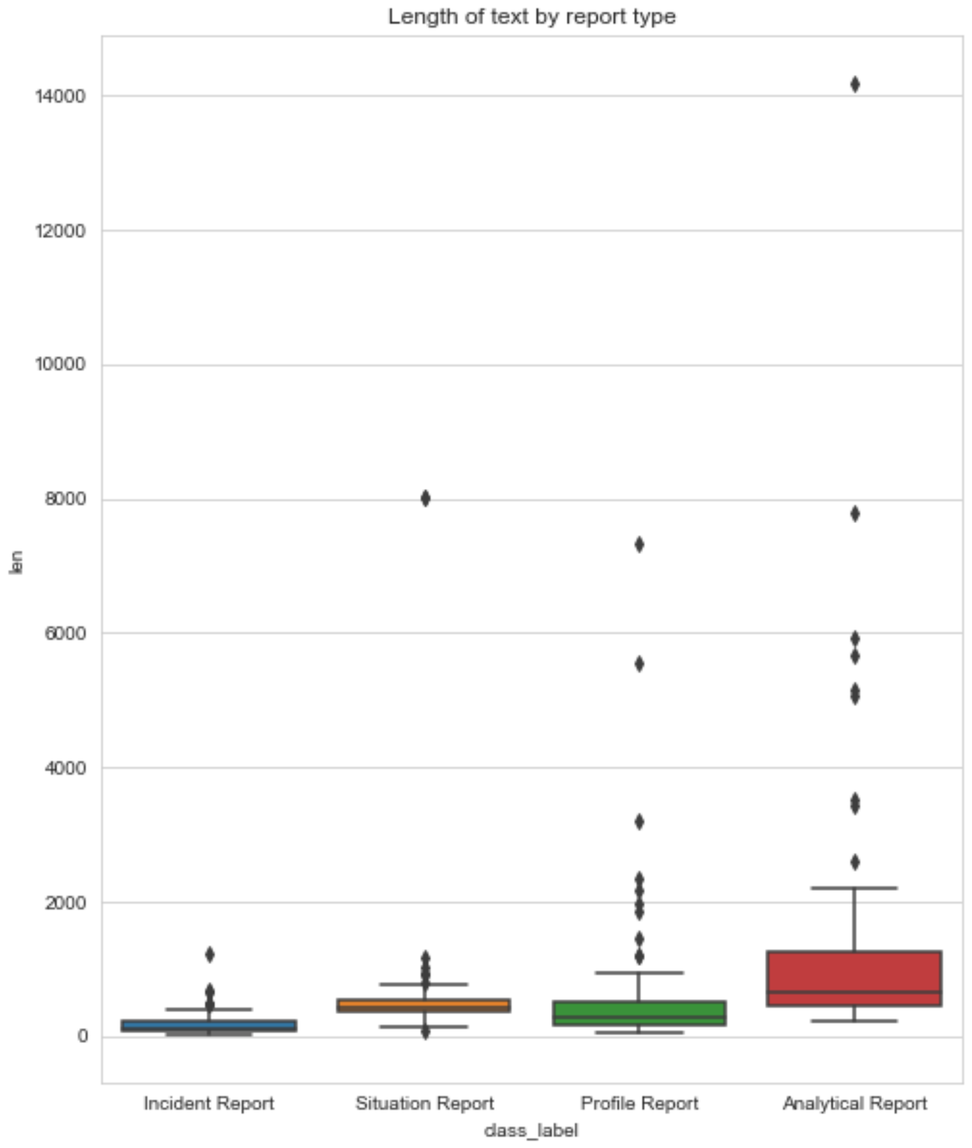
Traditional NLP

Exploratory process - charts



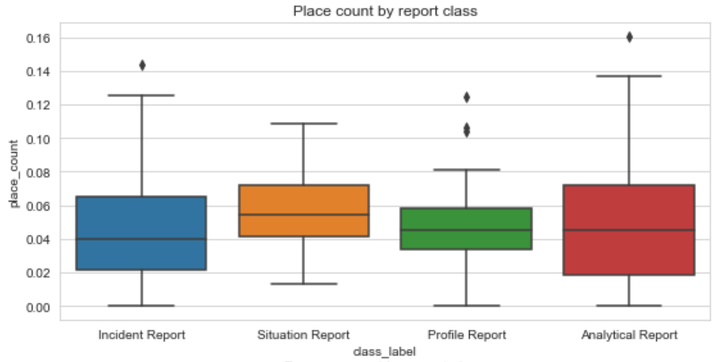
Traditional NLP

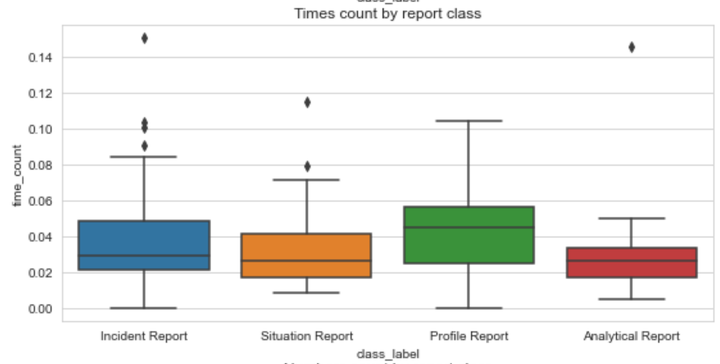
Exploratory process - charts

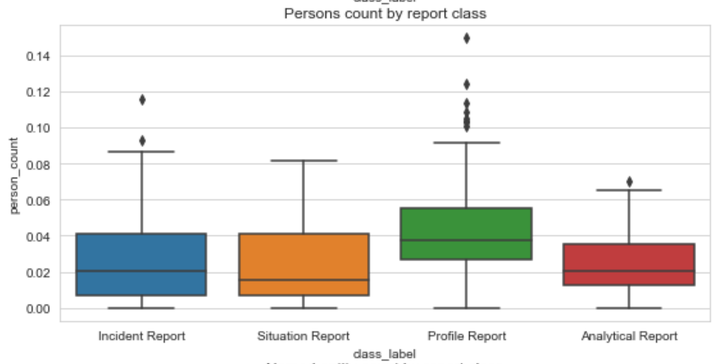


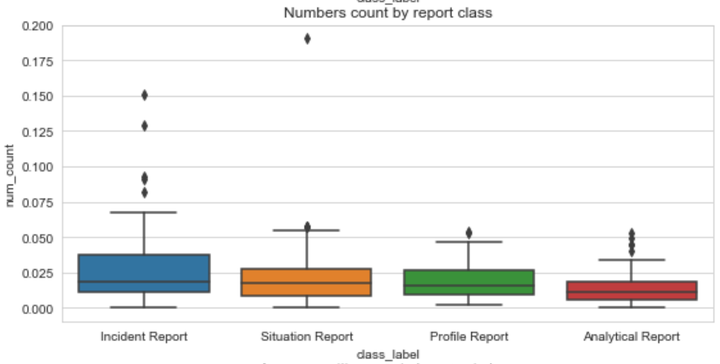
Traditional NLP

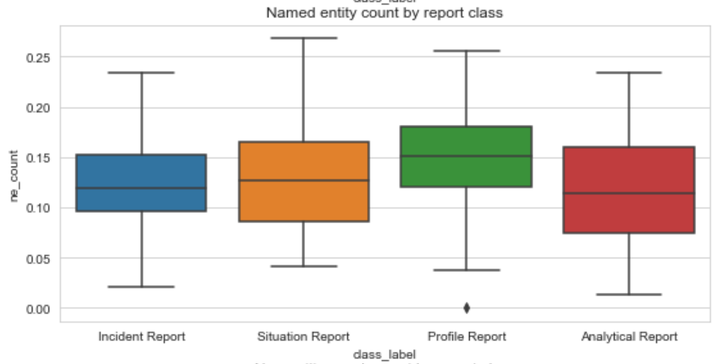
Exploratory process - charts









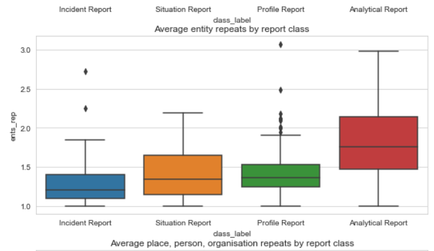


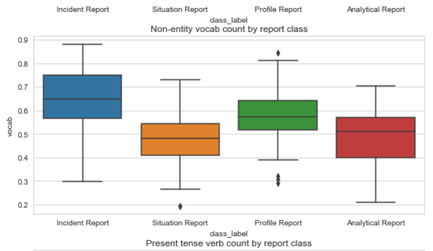
Average entity repeats by report class

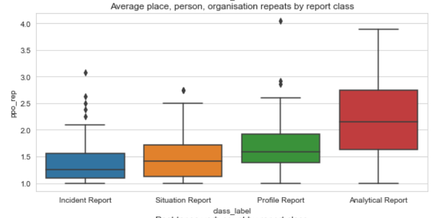
Non-entity vocab count by report class

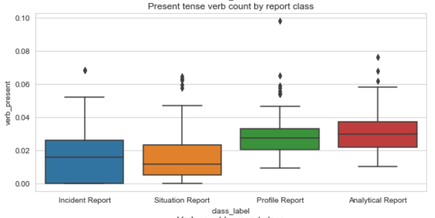
Traditional NLP

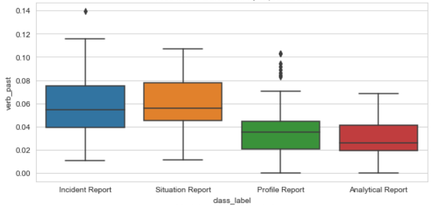
Exploratory process - charts

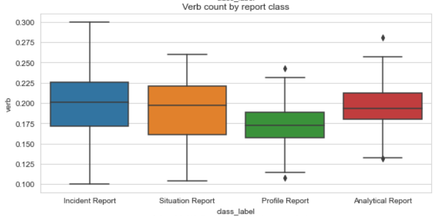


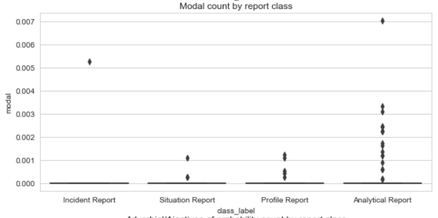


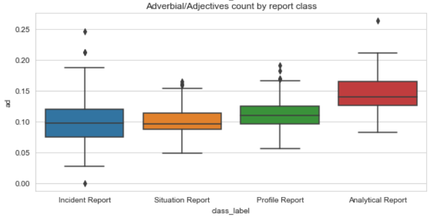


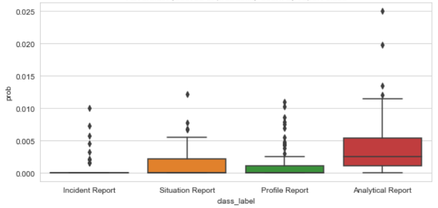


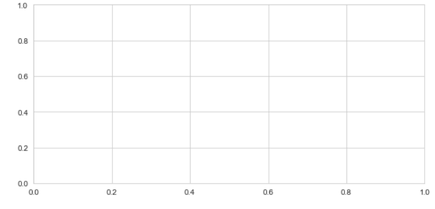












Traditional NLP

Iterative process: one

- Document lengths, normalised counts of organisations, places, times, persons, numbers.

```
Train:                    precision    recall  f1-score   support

  Incident Report       0.97      0.98      0.97        59
Situation Report        0.96      0.99      0.97        70
   Profile report       0.99      0.99      0.99        80
Analytical report       0.99      0.95      0.97        79

         accuracy                           0.98       288
        macro avg       0.97      0.98      0.98       288
     weighted avg       0.98      0.98      0.98       288

Validation:                    precision    recall  f1-score   support

  Incident Report       0.67      0.67      0.67        15
Situation Report        0.76      0.72      0.74        18
   Profile report       0.68      0.75      0.71        20
Analytical report       0.84      0.80      0.82        20

         accuracy                           0.74        73
        macro avg       0.74      0.73      0.74        73
     weighted avg       0.74      0.74      0.74        73

[[10  1  1  3]
 [ 0 13  5  0]
 [ 2  3 15  0]
 [ 3  0  1 16]]
```

```
26 misclassified
          class_label         pred_label
2      Incident Report  Analytical Report
17    Situation Report     Profile Report
24    Situation Report    Incident Report
53   Analytical Report   Situation Report
54   Analytical Report     Profile Report
57   Analytical Report   Situation Report
81    Situation Report  Analytical Report
96      Profile Report  Analytical Report
101     Profile Report  Analytical Report
109     Profile Report  Analytical Report
140     Profile Report    Incident Report
148     Profile Report    Incident Report
164     Profile Report    Incident Report
183  Analytical Report   Situation Report
202  Analytical Report   Situation Report
212    Incident Report     Profile Report
216    Incident Report     Profile Report
234    Incident Report     Profile Report
248    Incident Report     Profile Report
259    Incident Report     Profile Report
267   Situation Report    Incident Report
273   Situation Report     Profile Report
281   Situation Report  Analytical Report
307   Situation Report  Analytical Report
329   Situation Report    Incident Report
337  Analytical Report    Incident Report
```

Traditional NLP

Iterative process: two

- Average repeats of entities, places people and organisations, normalised vocabulary (non-entity) count

```
Train:                    precision    recall  f1-score   support

  Incident Report       0.98      1.00      0.99        59
Situation Report        0.99      1.00      0.99        70
  Profile report        1.00      1.00      1.00        80
Analytical report       1.00      0.97      0.99        79

         accuracy                           0.99       288
        macro avg       0.99      0.99      0.99       288
     weighted avg       0.99      0.99      0.99       288

Validation:                    precision    recall  f1-score   support

  Incident Report       0.83      0.67      0.74        15
Situation Report        0.72      0.72      0.72        18
  Profile report        0.73      0.80      0.76        20
Analytical report       0.90      0.95      0.93        20

         accuracy                           0.79        73
        macro avg       0.80      0.78      0.79        73
     weighted avg       0.80      0.79      0.79        73

[[10  2  2  1]
 [ 0 13  4  1]
 [ 1  3 16  0]
 [ 1  0  0 19]]
```

```
17 misclassified
           class_label         pred_label
24    Situation Report    Incident Report
48   Analytical Report    Incident Report
53   Analytical Report     Profile Report
75    Situation Report  Analytical Report
81    Situation Report  Analytical Report
86     Incident Report   Situation Report
96      Profile Report  Analytical Report
140     Profile Report    Incident Report
148     Profile Report    Incident Report
164     Profile Report    Incident Report
178  Analytical Report     Profile Report
202  Analytical Report   Situation Report
212    Incident Report     Profile Report
234    Incident Report     Profile Report
248    Incident Report     Profile Report
259    Incident Report     Profile Report
337  Analytical Report    Incident Report
```

Traditional NLP

Iterative process: three

- Normalised counts of present tense verb, past tense verb, verb, modal auxiliary, adjectives/adverbs

```
Train:                    precision    recall  f1-score   support

  Incident Report       1.00      1.00      1.00        59
Situation Report        0.99      1.00      0.99        70
  Profile report        1.00      1.00      1.00        80
Analytical report       1.00      0.99      0.99        79

        accuracy                            1.00       288
       macro avg        1.00      1.00      1.00       288
    weighted avg        1.00      1.00      1.00       288

Validation:                   precision    recall  f1-score   support

  Incident Report       0.71      0.80      0.75        15
Situation Report        0.89      0.94      0.92        18
  Profile report        0.85      0.85      0.85        20
Analytical report       0.88      0.75      0.81        20

        accuracy                            0.84        73
       macro avg        0.83      0.84      0.83        73
    weighted avg        0.84      0.84      0.84        73

[[12  1  0  2]
 [ 0 17  1  0]
 [ 2  1 17  0]
 [ 3  0  2 15]]
```

```
13 misclassified
           class_label         pred_label
48   Analytical Report   Situation Report
54   Analytical Report   Situation Report
77    Situation Report     Profile Report
81    Situation Report  Analytical Report
96      Profile Report  Analytical Report
109     Profile Report  Analytical Report
148     Profile Report    Incident Report
234    Incident Report     Profile Report
273   Situation Report  Analytical Report
281   Situation Report  Analytical Report
323   Situation Report     Profile Report
329   Situation Report    Incident Report
337  Analytical Report    Incident Report
```

Traditional NLP

Iterative process: four

- Normalised count of adverbials and adjectives of probability:

  - Normalised count of adverbials and adjectives of probability: 'prob'

```
Train:                    precision    recall  f1-score   support

  Incident Report       1.00      1.00      1.00        59
Situation Report        0.99      1.00      0.99        70
   Profile report       1.00      1.00      1.00        80
Analytical report       1.00      0.99      0.99        79

         accuracy                           1.00       288
        macro avg       1.00      1.00      1.00       288
     weighted avg       1.00      1.00      1.00       288

Validation:                   precision    recall  f1-score   support

  Incident Report       0.81      0.87      0.84        15
Situation Report        0.94      0.89      0.91        18
   Profile report       0.78      0.90      0.84        20
Analytical report       0.94      0.80      0.86        20

         accuracy                           0.86        73
        macro avg       0.87      0.86      0.86        73
     weighted avg       0.87      0.86      0.86        73

[[13  0  1  1]
 [ 0 16  2  0]
 [ 1  1 18  0]
 [ 2  0  2 16]]
```

```
 11 misclassified
          class_label         pred_label
48   Analytical Report   Situation Report
81    Situation Report  Analytical Report
96      Profile Report  Analytical Report
148     Profile Report    Incident Report
234    Incident Report     Profile Report
259    Incident Report     Profile Report
273   Situation Report  Analytical Report
281   Situation Report     Profile Report
323   Situation Report     Profile Report
329   Situation Report    Incident Report
337  Analytical Report     Profile Report
```

Traditional NLP

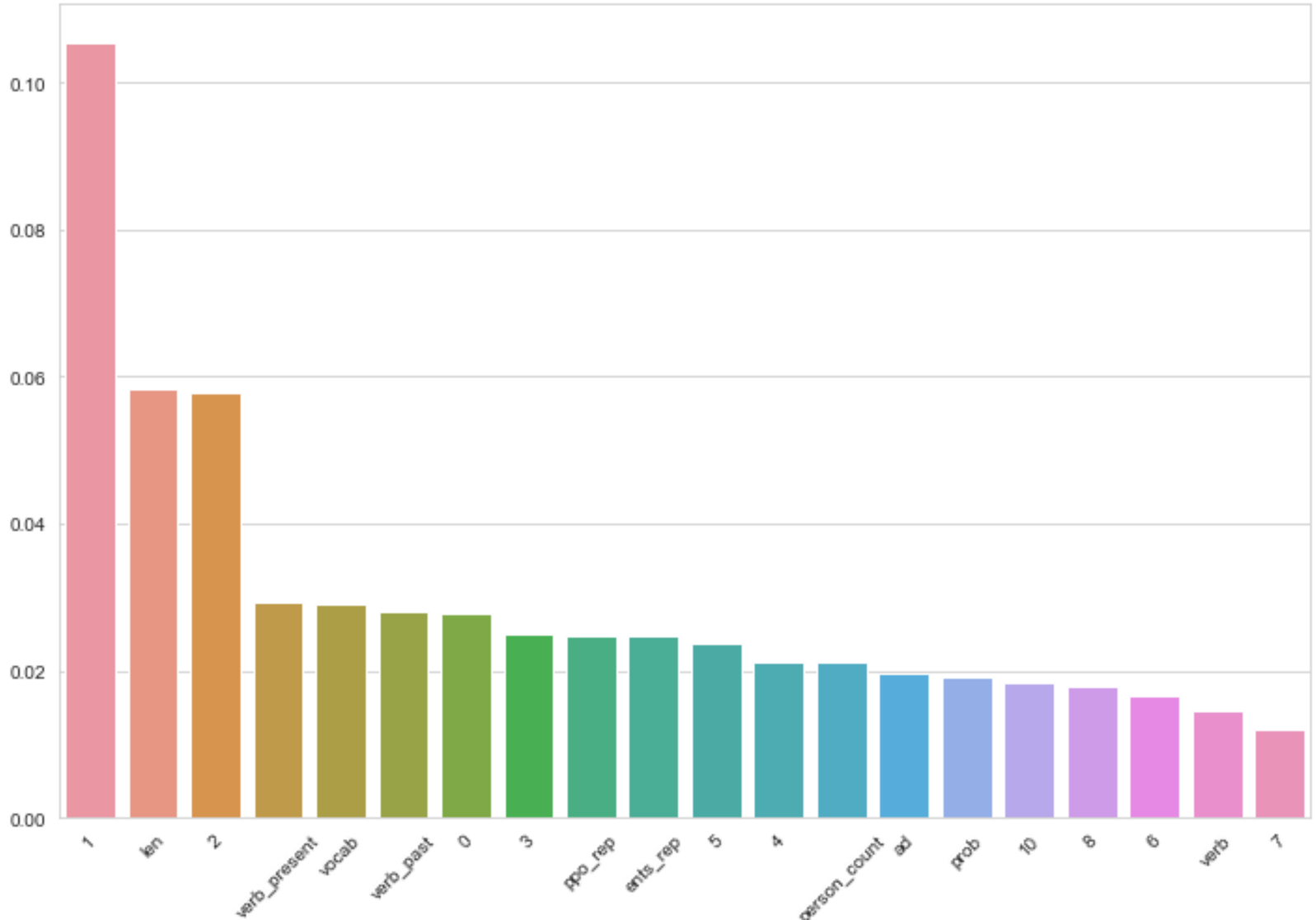
Iterative process: five

- Adding SVD (100) of TF-IDF of lemmatised documents.

```
Train:                    precision    recall  f1-score   support

  Incident Report       1.00      1.00      1.00        59
 Situation Report       1.00      1.00      1.00        70
   Profile report       1.00      1.00      1.00        80
Analytical report       1.00      1.00      1.00        79

         accuracy                           1.00       288
        macro avg       1.00      1.00      1.00       288
     weighted avg       1.00      1.00      1.00       288

Validation:                     precision    recall  f1-score   support

  Incident Report       0.79      0.73      0.76        15
 Situation Report       0.94      0.94      0.94        18
   Profile report       0.82      0.90      0.86        20
Analytical report       0.95      0.90      0.92        20

         accuracy                           0.88        73
        macro avg       0.87      0.87      0.87        73
     weighted avg       0.88      0.88      0.88        73

[[11  1  3  0]
 [ 0 17  0  1]
 [ 2  0 18  0]
 [ 1  0  1 18]]
```

Traditional NLP

Feature importance



Traditional NLP
Finalise model

- Random search cross validation was conducted with random forest (Best parameters: {’n_estimators': 1000, 'min_samples_split': 2, 'min_samples_leaf': 1, 'max_depth': 8, 'bootstrap': True}) and CatBoost (Best parameters: {‘learning_rate': 0.3, 'l2_leaf_reg': 15, 'iterations': 500, 'depth': 7}).
- Both yielded similar results: F1 test score, random forest: 0.91, CatBoost: 0.91.
- The parameter space could be shifted, expanded and or zoomed in to allow for better results. However, within the given time, the result is satisfactory.
- I chose random forest over gradient boost machine because of similar results and much faster run time for random forest

BERT

BERT

What’s BERT

- BERT(Bidirectional Encoder Representations from Transformer) is semi-supervised in MLM and NSP, pretrained on Wikipedia and BooksCorpus.
- It can be used on downstream tasks such as sequence classification, question-answering, etc.

BERT

More challenges and ideas

- Challenge 1: the documents are often a lot longer than BERT’s maximum length 510.
- Solution 1: Simple truncate (head-only, tail-only, head-and-tail, hierachical)
- Solution 2: Longformer
- Challenge 2: BERT might need more labelled data (subsequently proved untrue!).
- Solution 1: Make more data by dividing documents into segments, each labelled accordingly.
- Solution 2: Back-translate to create more data
- Solution 3: Fine tune on semi-supervised task before fine tune again with specific task. MLM -> Classification

BERT

What I tried

- Simple truncate head-only (There is a paper demonstrating complicated methods (hierarchical) do not work as well as simple truncate.
- Make more data by dividing documents into segments, each labelled accordingly. The result was disappointing, so this approach was abandoned.
- Back-translate to create more data. (Tried towards the end, but couldn’t resolve an vocabulary error)
- Fine tune on a semi-supervised task. Wasn’t able to obtain more unlabelled data, but then discovered RoBERTa is actually a great alternative pretrained on news and online articles!

BERT

Result - BERT

- BERT produced comparable results to traditional NLP, after training for a couple of hours on a AWS EC2 instance

	precision	recall	f1-score	support
Incident Report	0.92	0.97	0.94	88
Situation Report	0.97	0.89	0.93	99
Profile report	0.97	0.97	0.97	100
Analytical report	0.91	0.96	0.93	74
accuracy			0.94	361
macro avg	0.94	0.95	0.94	361
weighted avg	0.95	0.94	0.94	361

```
[[85  2  0  1]
 [ 7 88  1  3]
 [ 0  0 97  3]
 [ 0  1  2 71]]
```

BERT

What’s RoBERTa

- RoBERTa builds on BERT, but has these key differences:
- Bite Piece Encoding (BPE) in stead of Word Piece Encoding (WPE)
- Larger mini-batches, higher learning rates and longer sequences.
- No training on NSP task
- Dynamic maskin pattern
- Trained on much bigger corpus, Wikipedia, BookCorpus, plus CC-NEWS, OpenWebText and STORIES. You can see it as having already pertained on similar corpus to our training data.

BERT

Result - RoBERTa

- RoBERTa produces even better results than both traditional methods and BERT.

	precision	recall	f1-score	support
Incident Report	0.95	1.00	0.97	18
Situation Report	1.00	0.85	0.92	20
Profile report	1.00	1.00	1.00	20
Analytical report	0.88	1.00	0.94	15
accuracy			0.96	73
macro avg	0.96	0.96	0.96	73
weighted avg	0.96	0.96	0.96	73

```
[[18  0  0  0]
 [ 1 17  0  2]
 [ 0  0 20  0]
 [ 0  0  0 15]]
```

Final thoughts

Do you want to cook?

OR

Final thoughts
Traditional NLP

- Reading misclassified documents can help building intuition and crafting features.
- For deployment, I would be very happy to deploy a traditional NLP model into production, especially when speed is important, because random forest took so little time to train and has great result and interpretability.
- However, I am also aware the small size of labeled data which could perhaps make my feature engineering overfit to the training data.
- Also to consider model maintenance: new data may need new feature engineering. Inference requires preparation: tokenisation, feature extraction, and it won't be instant.
- A bit like making pizza. If the model (pizza making process) has only been exposed to a certain type of data (i.e. ready-to-go topping) on it, when new kind of data comes in (i.e. raw peppers), you might need to update your preparation method - new feature engineering (roasting before cooking).
- What about a language you aren't familiar with? How do you feature engineer?

Final thoughts
BERT and RoBERTa

- RoBERTa outperforms BERT by a large margin and takes similar amount of time to fine tune, thus preferable if BERT based model is chosen.
- For the benefit of downstream tasks that utilises BERT (classification, question-answering, entity recognition, etc.), it is better to fine tune BERT parameters on domain specific corpus on semi-supervised learning tasks, i.e. MLM, before task specific models.

Final thoughts
BERT and RoBERTa

- This is much like making sour dough or Japanese ramen soup. You have a ‘mother’ BERT (or spin-offs) model, which is fine tuned as new unlabelled data comes in, on semi-supervised tasks to keep it ‘up-to-date’. This update will increase accuracy of BERT-based models (children) on all downstream tasks.
- This will reduce training time and the amount of labelled data required for all downstream tasks.
- A bit like sour dough or ramen soup. All your food made with mother dough or soup taste delicious because the base is already delicious and constantly updated to the new ingredients (data).