# CS 839 - Project Stage 1

Pratyush Mahapatra, Akshata Bhat, Felipe Gutierrez Barragan

March 11, 2019

## 1 Annotation Statistics

- **Entity Type:** Any location that is a city, region, country, or continent.
- **Annotation Tags:** <loc> ... </loc>.
- **Example annotated entity:** "<loc>WASHINGTON</loc>—<loc>U.S.</loc> industrial output fell...". For locations where two as well as one word act as locations, we tag it as follows : <loc>NEW <loc>DELHI</loc></loc>
- **Number of annotated entities:** 3486.
- **Set $I$:** You can find set $I$ in this link: Browsable Set I Directory.
  - Number of documents: 220.
  - Number of annotated entities before pre-processing rules (positive examples): 2,180.
  - Number of annotated entities after pre-processing rules (positive examples): 2,164.
  - Number of negative examples before pre-processing rules: 226,983.
  - Number of negative examples after pre-processing rules: 29,728.
- **Set $J$:** You can find set $J$ in this link: Browsable Set J Directory.
  - Number of documents: 110.
  - Number of annotated entities before pre-processing rules (positive examples): 1,306.
  - Number of annotated entities after pre-processing rules (positive examples): 1,283.
  - Number of negative examples before pre-processing rules: 137,574.
  - Number of negative examples after pre-processing rules: 19,287.

## 2 Results with Learned Model

We apply some simple pre-processing rules to the dataset to reduce the skewness of the dataset. Please refer to section 4 for a brief description of the rules applied. During this step a few false negatives are introduced as seen in the numbers reported in section 1. The number of false negatives is negligible and does not significantly impact the results presented hereafter.

**Cross-validation results on set $I$ (First Time):** Table 1 has the results for 10-fold cross-validation for the first time around. Since linear regression is not often used for classification we omit those results. The models evaluated in this table contained a small set of feautures that we initially thought were relevant. Random Forest was the best overall classifier.

**Cross-validation results on set $I$ (Final):** Table 2 has the results for 10-fold cross-validation. We debug our model by analyzing false positive and negatives, and add features accordingly. Random Forest continued to be the best overall classifier (classifier X).

| Model | Mean Accuracy | Mean Precision | Mean Recall | Mean F1 Score |
|---|---|---|---|---|
| Logistic Regression | 0.932 | 0.284 | 0.099 | 0.147 |
| Decision Tree | 0.937 | 0.443 | 0.430 | 0.435 |
| **Random Forest** | 0.947 | 0.618 | 0.397 | 0.481 |
| SVM | 0.932 | 0.000 | 0.000 | 0.000 |

Table 1: **Cross-validation Results First Time.**

| Model | Mean Accuracy | Mean Precision | Mean Recall | Mean F1 Score |
|---|---|---|---|---|
| Logistic Regression | 0.941 | 0.653 | 0.267 | 0.367 |
| Decision Tree | 0.953 | 0.629 | 0.616 | 0.622 |
| **Random Forest** (X) | 0.960 | 0.795 | 0.531 | 0.634 |
| SVM | 0.930 | 0.200 | 0.003 | 0.006 |

Table 2: **Cross-validation Results Final.**

**Test results on set $J$:** We use the Random Forest model we obtained from the cross-validation step. We train it on the full set I and test on set J. The same pre-processing rules are applied to both.

1. **Accuracy:** 0.964
2. **Precision:** 0.764
3. **Recall:** 0.612
4. **F1-Score:** 0.680

## 3 Results with Learned Model and Rule-based Post-processing

Please refer to section 5 for details on the post-processing rules used.

1. **Accuracy:** 0.973
2. **Precision:** 0.853
3. **Recall:** 0.699
4. **F1 Score:** 0.768

## 4 Pre-processing Rules

We apply the following pre-processing on the train and test set to reduce the skewness of the dataset. The skewness of the dataset can be easily seen in the numbers reported in section 1.

- Only keep samples with at least one capital letter.

- Remove all samples that have digits in them.
- Remove all samples that have string length of 1.
- Remove all samples that have string length of 2 and whose letters are lower case. This case we ignore entries such as "in" but do not miss locations such as "US".
- Remove all samples that are equal to the following: "A", "An", "The", "And".
- Remove all samples that where the string we are looking at crosses a sentence. For instance "China. He" would be removed. We make sure that locations such as U.S. do not get removed.
- For all samples composed of strings with 2 words in them, check if the first word has a non-alphabetical character. If it does remove that sample. For example : "China's population"

The above pre-processing rules can be found in preprocessing.py.

## 5 Post-processing Rules

The rules derived in this section are obtained by splitting set I into train (Set P) and validation sets (Set Q). We then analyze the false positives and false negatives obtained when testing on the validation set.

We apply the following post-processing rules. All post-processing rules are applied in the post_processing function in ner_model.py.

- If word contains the "$" character classify it as negative. We often found that whenever currency was mentioned our model would classify as positive.
- If word is any month of the year (e.g. January, February, etc) classify as negative.

**Whitelist:** The following are common locations that we always want to classify as positive.

- US
- U.S.
- UK
- EU
- INDIA
- CHINA
- GERMANY

**Blacklist:** Our model easily mistakes people's names and organizations for locations. We included the following words in our blacklist

- Trump
- African, European, American, Asian

- Bank
- RBI
- House
- Treasure
- Union
- Brexit
- New
- Kingdom

All post-processing rules can be found in this file dictFeatures.py

## 6 Discussion

We were able to achieve a precision of 85% but we found that we were not able to push it further despite having 151 features. The reason for that is as follows:

1. Our model confuses between Names, Organizations and Locations. This happens because the grammatical construction for all three has high similarity
2. Missing annotations - Due to the length of each article in our data set, we ended up missing annotations in our initial phase and we had to keep revising our data set throughout the project.
3. Huge data set- We had an extremely varied and big data set. This made identifying all the entities a much bigger challenge. This coupled with correcting annotations throughout the project made it difficult to achieve the desired goal

We believe that we could have overcome the first drawback by implementing more complex features that capture context. We also believe that our precision would have increased if we would have used dictionary based features (like Bag of Words). However, we decided to not include such features because we thought they would simply memorize the locations. In retrospect, some memorization might be needed in situations where the sentence structure is very similar for different entities, as it is in our case (organizations/locations/names).

## 7 Takeaways/Learnings

- Selection of data set: We learnt, albeit late, the importance of choosing a data set of an appropriate size. Reduces the time and labour cost of cleaning the data.
- Making use of context while annotating : We faced many instances of locations where the context was different. For example, "US Federal Reserve Bank..." In our data set, we annotated all instances of the occurrences of US, but using context and annotating only when the country was explicitly called for would have helped.