EVALUATION REPORT

EVENT IDENTIFICATION, DETECTION AND SUMMARIZATION

BY

SRI HARSHA KESAPRAGADA

TANNU PRIYA

Table of Contents:

1. Problem Definition

In this Milestone, we aim to present a precise evaluation of our system which does Event identification, detection and summarization. This system extracts and summarizes events occurred in three target countries about "Riots/Protests" and "Violence against civilians" by ingesting multiple input, such as Twitter, Google News, The Hindu, Times of India, ABC News, BBC News, sub local news etc. and would give detailed information about the events.

End result of this is system is a csv file that that provides details to the user in below format:

- Event date
- Location
- Event type
- Parties involved
- Detailed Summary of the event
- Data sources

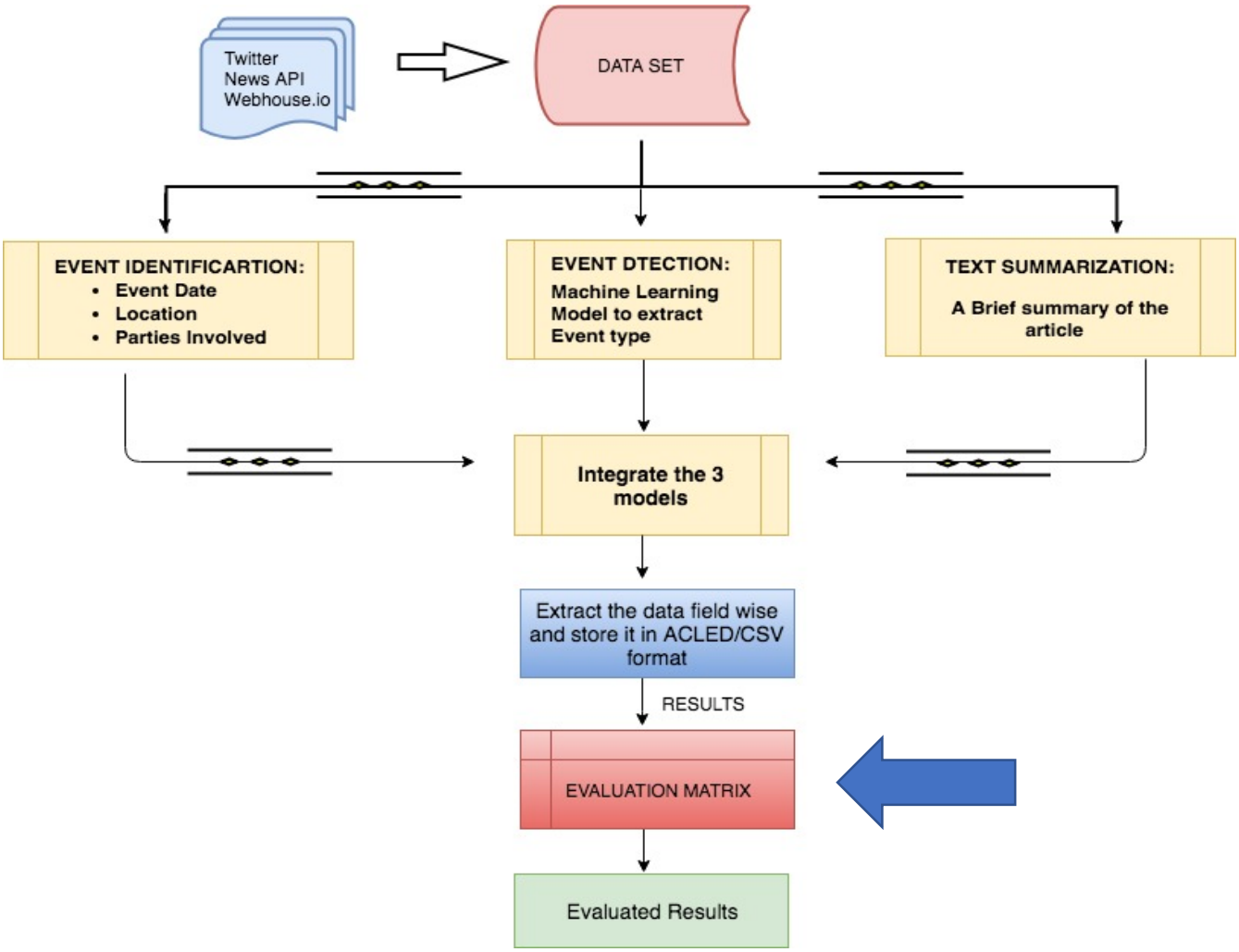
2. Data Sources:

We have used below Data sources for collecting news articles:

- The Times of India
- Hindustan Times (India)
- The Hindu
- DNA Daily News and Analysis
- Daily Excelsior
- Kashmir Monitor
- State Times (India)
- Hans India
- Herald Goa
- Indian Express
- Pakistan Press International
- Odisha Sun Times
- Twitter
- News API
- Webhose.io

For evaluating our system, we manually collected a set of around 100 articles which ACLED summarized under Riots/Protests or Violence against Civilians category. These articles were collected over a span of one month. We fed these articles as input to our system and then we evaluated the performance of our model through different metrics for each individual column.

3. Solution Architecture :



We have used different metrics for the evaluation of 3 Tasks, described below:

- Classification/ Event Detection:
    - Precision and Recall using Confusion Matrix
- Text Summarization:
    - ROUGE-N: N-gram Co-Occurrence
    - ROUGE-L: Longest Common Subsequence
- Event Identification:
    - F1-score by NEREVAL

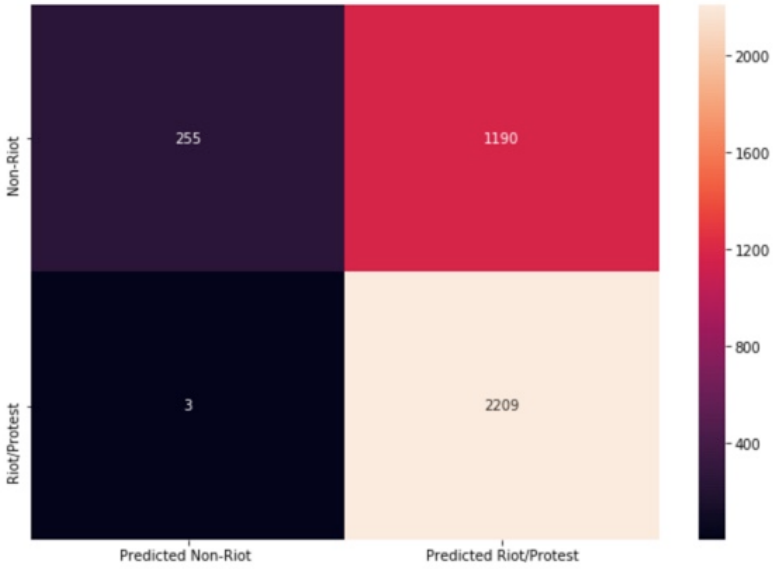
PART 1: TASK: Classification/ Event Detection:

Evaluation Metric Used: Precision and Recall using Confusion Matrix

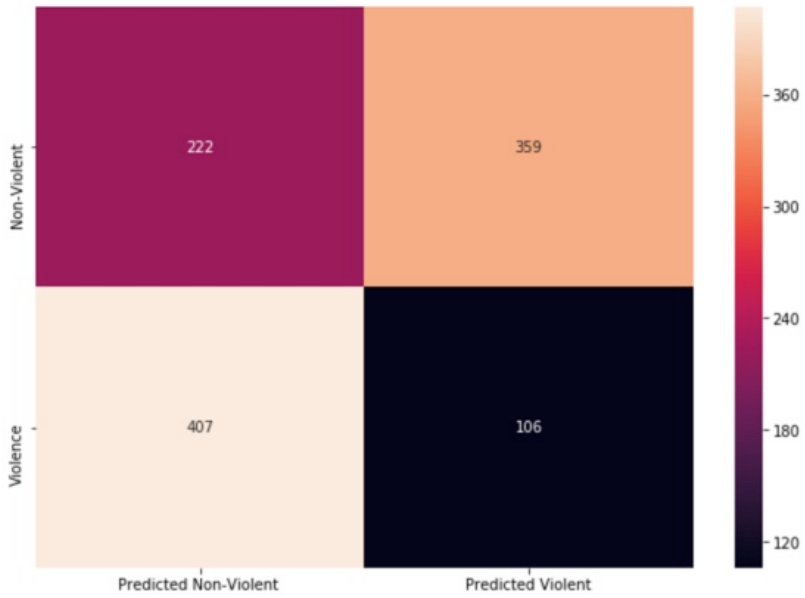
For Event Classification, we have implemented a One vs Rest Classification model where we have trained two classifiers, one for Riots/Protests and the other for Violence against Civilians. For each classifier, we have used data from ACLED as training data with samples of that class as positive samples and all other samples as negative samples.

We extracted the text from the articles and used our classifiers to categorize each and every article. If a classifier tags an article as negative, it will be fed to the next classifier. We mark an article as negative if both the classifiers mark it as negative.

Confusion Matrix for Training Phase:



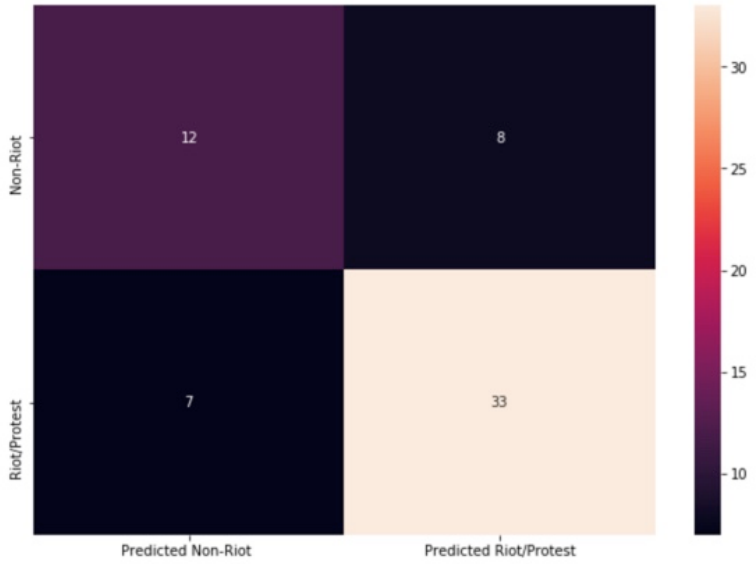
Visualization for Training Accuracy of Riots/Protests Classifier



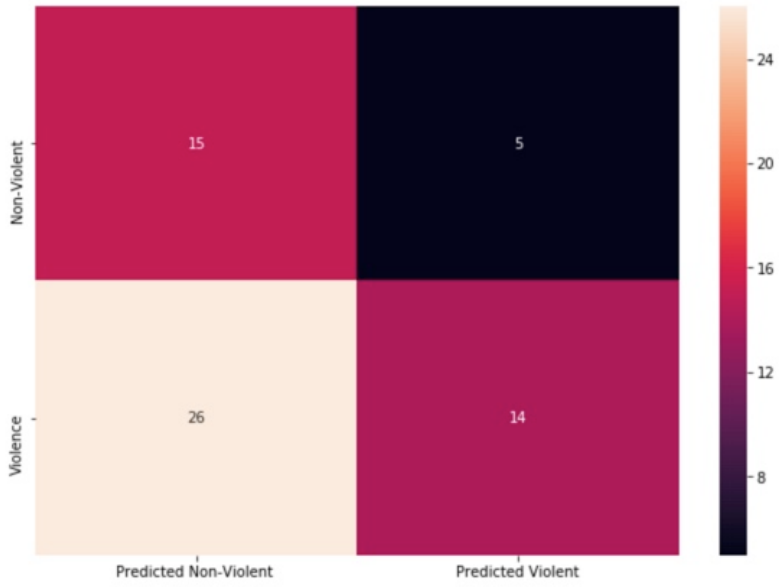
Visualization for Training Accuracy of Violence Against Civilians Classifier

Since ACLED has more data for Riots/Protests we got a very good training accuracy for our Riots/Protest Classifier. We got an **average-precision recall score of 0.65** when compared to the **average-precision recall score of 0.35** for Violence classifier.

Confusion Matrix for Evaluation Phase:



Visualization for Evaluation Accuracy of Riots/Protests Classifier



Visualization for Training Accuracy of Violence Against Civilians Classifier

As mentioned before the evaluation accuracy of the violence classifier is less due to less number of samples from ACLED whereas the Riots/Protests Classifier predicted almost all the true riot articles with a precision score of 70%.

PART 2: Task: Text Summarization.

Evaluation Matric Used:

- ROUGE-N: N-gram Co-Occurrence
- ROUGE-L: Longest Common Subsequence

ROUGE stands for Recall-Oriented Understudy for Gisting Evaluation. ROUGE automatically determine the quality of a summary by comparing it to ACLED summaries created by humans(Reference). The measures count the number of overlapping units such as n-gram, word sequences, and word pairs between our system generated summary and the ideal summaries created by ACLED.

1)ROUGE-N: N-gram Co-Occurrence:

We are taking computing N-gram (here, n=1) overlaps for every summary and then we are computing average of all the individual ROUGE SCORES to gauge how well the text summarizer is working.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

Where n stands for the length of the n-gram, is the maximum number of n-grams co-occurring in a candidate summary and a set of reference summaries. We are considering n =1 to evaluate summaries.

2) ROUGE-L: Longest Common Subsequence:

In Longest common subsequence we are taking into account sentence level structure similarity and it identifies longest co-occurring in sequence n-grams automatically.

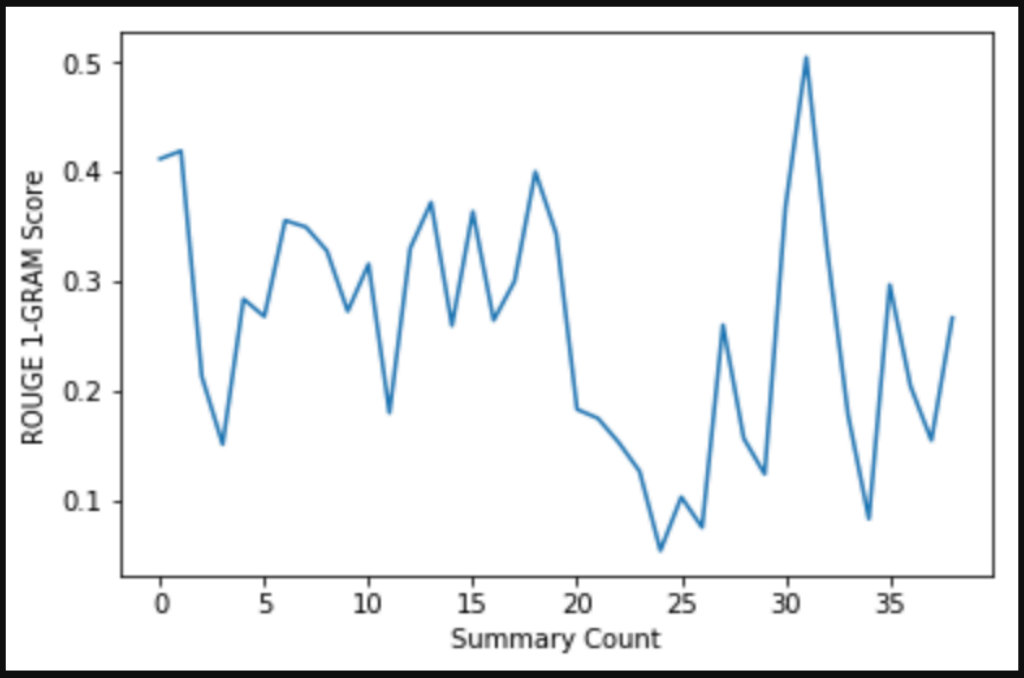
Results:

We have computed average of individual summaries to gauge the system.

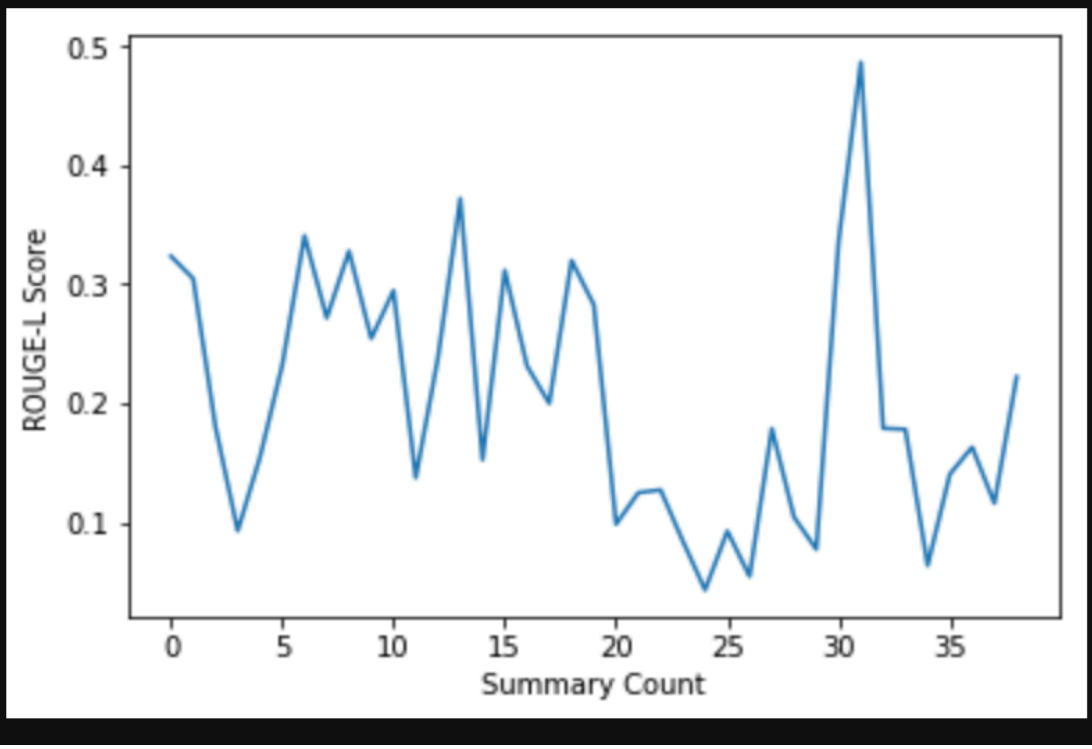
```
sum_of_rouge_bi_gram_score = 0
sum_of_rouge_L_score = 0
avg_score_bi_gram = 0
avg_score_rouge_l = 0

for i in range (0, len(ref_data)):
    #COMPUTING ROUGE N-GRAM SCORE
    rouge_1 = rouge.rouge_n(summary = summary_data[i], references = ref_data[i], n=1)
    sum_of_rouge_bi_gram_score += rouge_1
    #COMPUTING ROUGE-L SCORE:
    rouge_l = rouge.rouge_l(summary = summary_data[i],references = ref_data[i])
    sum_of_rouge_L_score += rouge_l
#CALCULATING AVG
avg_score_bi = (sum_of_rouge_bi_gram_score/(len(ref_data)))
avg_score_rouge_l = (sum_of_rouge_L_score/(len(ref_data)))
print("ROUGE BI Gram Score: {}, ROUGE-L Score: {} ".format(rouge_1, rouge_l).replace(", ", "\n"))
```

```
ROUGE BI Gram Score: 0.26666666666666666
ROUGE-L Score: 0.2222222222222222
```

```
AVERAGE ROUGE 1-Gram Score: 0.26666666666666666
Text(0, 0.5, 'ROUGE 1-GRAM Score')
```

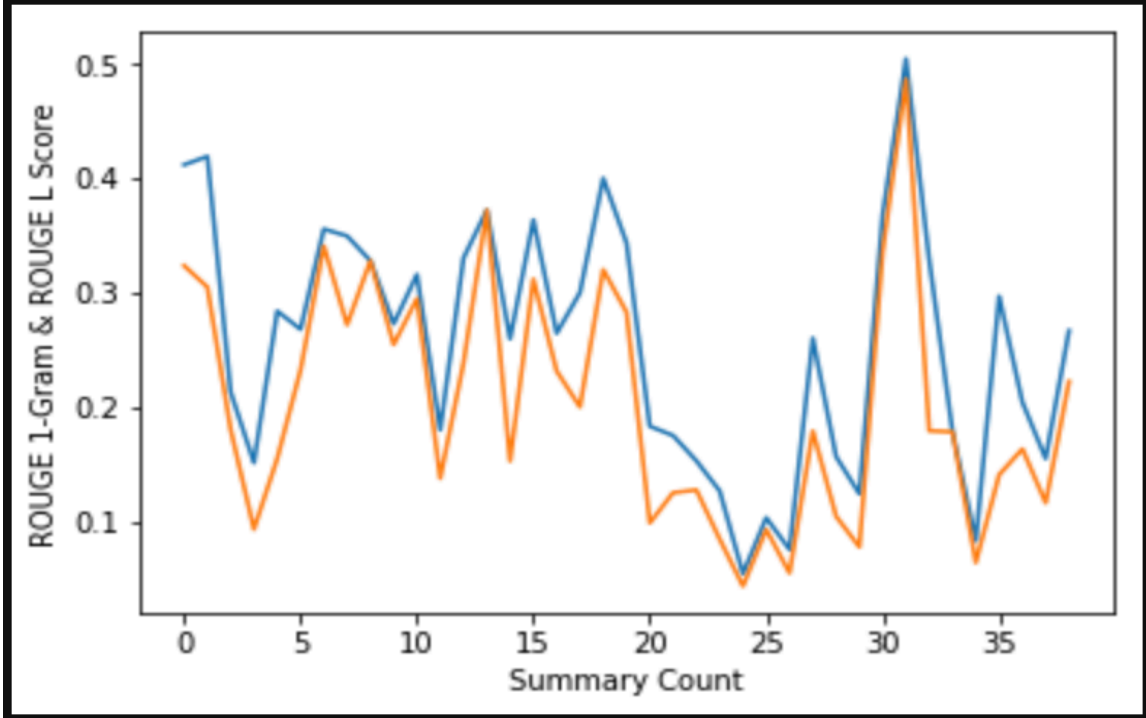



```
AVERAGE ROUGE-L Score: 0.2222222222222222
Text(0, 0.5, 'ROUGE-L Score')
```



COMBINED EVALUATION:

```
ROUGE BI Gram Score: 0.26666666666666666
ROUGE-L Score: 0.2222222222222222
Text(0, 0.5, 'ROUGE 1-Gram & ROUGE L Score')
```



PART 3: ENTITY IDENTIFICATION:

We used a tool called NEREVAL which essentially is a script to evaluate named entity recognition systems based on F1 score. It evaluates an NER system according to two axes: whether it is able to assign the right type to an entity, and whether it finds the exact entity boundaries. For both axes, the number of correct predictions (COR), the number of actual predictions (ACT) and the number of possible predictions (POS) are computed. From these statistics, precision and recall can be derived:

$$\text{precision} = \text{COR}/\text{ACT}$$
$$\text{recall} = \text{COR}/\text{POS}$$

We used a tool called **SPACY** to extract the entities out of the text. To get the correct event date, we used **Stanford Temporal Tagger** which is a library for normalizing and recognizing time expressions. We used the article published date as reference date which is passed as a parameter to the tagger which then extracts and gives the correct date on which the article says a particular event has happened.

We used F1-score to evaluate the following three fields extracted by our system.

1. Location
2. Date
3. Parties Involved

We created a JSON object for every article tagged from the test set and created another JSON object using ACLED identified tags. We then used the nereval script to evaluate and we are able to get a **F1-score of 0.30**. This is due to the fact that our entity tagging depends on the quality of the summary generated by our summarizer and we aim to improve both the ROUGE score for summarization and the F1-score for entity identification.

PLANS FOR FUTURE:

Below are our plans to improve the accuracy for individual tasks:

1) Event classification:
   Since we are limited by the number of training samples from ACLED for a particular country, we are planning to incorporate more samples to train our classifiers using other conflict databases available online. Datasets like Uppsala Conflict Data Program(UDCP) and Urban Social Disorder Dataset(USD) have public datasets available and we are planning to use these datasets along with ACLED for our final evaluation.

2) Text Summarization:
   We are planning to optimize our summarization using Adapted TextRank[1] for Term Extraction which is a Generic Method of Improving Automatic Term Extraction Algorithms. Also, we plan to use BLEUCalculator for evaluation along with ROUGE N-gram and ROUGE-L.

3) Entity Identification:
   Our model dependents on the quality summary. Improving on Text summarization will help us improve data for entity Tagging.

[1] https://www.researchgate.net/publication/326904600_A_Generic_Method_of_Improving_Automatic_Term_Extraction_Algorithms

REFERENCES:

[1] C. yew Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.

[2] Angel X. Chang and Christopher D. Manning. 2012. SUTIME: A library for recognizing and normalizing time expressions. In LREC 2012.

[3] Steinberger, J., Ježek, K.: Evaluation measures for text summarization. Computing and Informatics 25, 1001–1025 (2012).