Refer to **scraper_A1.ipynb**

1b)

For my assignment, I scraped 138 articles about the Recent Delhi Elections from *the Hindu* website. I decided to use a single source instead of an aggregator like google news for several reasons. Firstly, many news sites require either a premium subscription or a login to access their sites, which is difficult to automate. Secondly, each news site has their own html format, which makes it difficult to scrape or leads to inconsistent data. Thirdly, different sources often report the same incident leading to redundant data. The Hindu website is well formatted with each article within an <articlebody> container

Here is how my scraper works:

I used Selenium with Undetected ChromeDriver, disabling the "AutomationControlled" feature to bypass anti-bot detection. Selenium also allowed me to log into my personal hindu account since there is a limit on the number of articles for non-premium users.

The Hindu luckily had a dedicated section about the Election, and I used BeautifulSoup to parse the links as well as iterating through multiple pages to collect all links. I then used Beautiful Soup to load each article, extracting the text from the **article body** from,while filtering out unwanted sections from certain repetitive class types such as ‘read more’ . Title and articles were stored into separate lists in order to avoid string formatting issues.

I then saved raw articles into a txt file, which allowed me to clean the article for boilerplate like photo sources and twitter posts, which were hard to automate. I also used **regular expressions** to remove any unwanted metadata. Finally I stored and exported the cleaned articles into a csv file for further analysis.

Refer to **pos_ner.ipynb** for code for 2.a and 2.b **and coref.ipynb** for 2.c while **2.d** was implemented on **entity_resolution.ipynb**. ***Resolved_entities.csv*** contains the dataset for all the above mentioned problems

2)d

For entity resolution, I first used a rule based step where I matched common abbreviations of entities that were common throughout the articles. For example AAP -> Aam Aadmi party and BJP -> Bharatiya Janata Party, EC-> Election Commission. Abbreviations are tough to match via string similarity algorithms so this was crucial. Spacy’s ner did not pick up any prefixes such as Mr. or Ms, However the post-fix “ji” was common which was removed.

I then used two entity resolution algorithms to match my entities. First, I used fuzzy Matching, which uses the Levenshtein distance to measure string similarities. This calculates the minimum number of single character edits required to change a word to another. If the similarity is above a threshold (85% in my case), it is a match. This is effective for correcting minor variations in spelling such as “Election Commission” and “Election Comission”, or “South Delhi” and “South-Delhi”.

Secondly, I used the Cosine Similarity to match words. It is calculated by dividing the dot product of the vectors by the product of their lengths. Therefore the cosine similarity depends on the angle rather than the magnitude of the vector. This is important for resolving entities with varying lengths. For example, “The Delhi Assembly polls” and “The Delhi polls”, would not be resolved using only fuzzing matching. It also helped match differences in full names such as “Parvesh Verma” and “Parvesh Sahib Singh Verma”, which were present in most articles. Using both algorithms allowed for more robust resolution for different problems.

3a) Used **ner_perf.ipynb**

I used label-studio to manually annotate the articles. The Spacy NER performed surprisingly well on the data set, with the following results.

```
True Positives: 1492
False Positives: 366
False Negatives: 454
Precision: 0.80
Recall: 0.77
F1 Score: 0.78
```

The performance of the NER model was calculated by comparing it to the manually annotated dataset which was treated as the gold standard. A true positive occurs if the NER model correctly predicts an entity label. A false negative occurs when the model fails to recognise an entity present in the ground truth dataset. A false positive occurs when the model makes a prediction which isn’t true.

Precision is calculated by dividing the true positive by True positive + False positives. This indicates how many of the entities that the model detected were actually correct. In this case 80% of all of Spacy’s labels were accurate. This number was also boosted by the fact that a significant number of entities in the articles were either names of Parties like AAP or Names of Candidates which appeared significantly high number of times and were almost always labelled correctly. Since the articles were about elections they had a lot of numbered entities which are relatively easy to predict.

Recall is calculated by dividing the True Positives by True positive + False negatives. It measures how many of the ground truth entities the model successfully detected. A recall of 77% means that the model missed about 23% of the correct entities. The spacy model was struggling to label laws and certain geographical locations. It also could not differentiate between “Delhi Assembly Elections” and “Delhi Assembly” labelling the former as a location.

The F1 score is the harmonic mean of both precision and Recall. An F1 score of 78% means that the model is well balanced between Precision and Recall and doesn’t overpredict or underpredict one way or the other.

3b) Refer to **Core_perf.ipynb**
Similarly Coreferences of 15 manually annotated datasets were compared to that of the Stanford CoreNLP model.

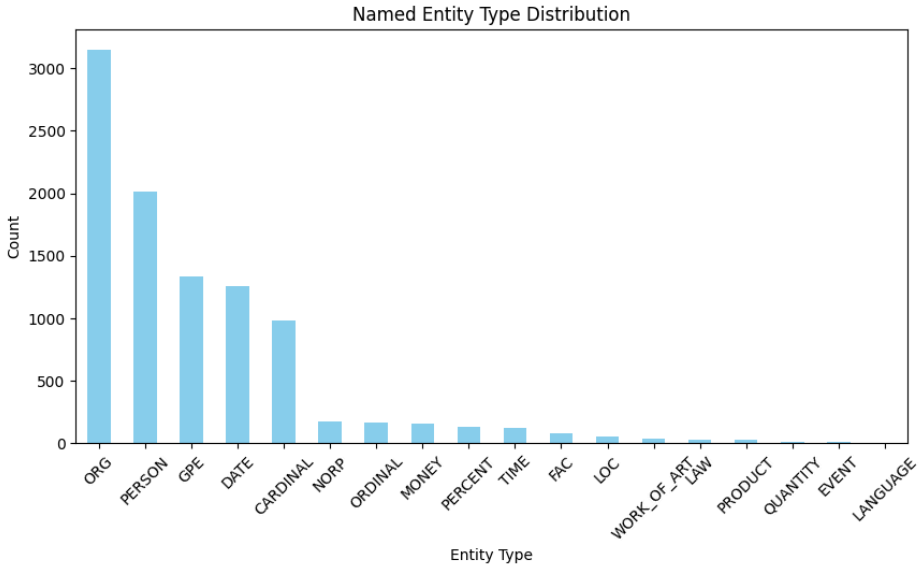
The results are as follows:

```
Precision: 0.4408
Recall: 0.4480
F1-score: 0.4183
Accuracy: 0.2728
```

The accuracy measures what percentage of the model's decisions were correct. The model had a fairly low precision of 27%. This means more than 70% of the models were incorrect. This was likely because the Stanford CoreNLP model did badly with long range dependencies.This was leading it to create multiple correlation chains of the same entity. It did much better in the example sentences which were only a few lines long.

Precision measures how many of the predicted coreference chains were correct. A 44% precision is comparatively low, suggesting that the model was clustering incorrect entities together. The model was also worse than Spacy in its NER, leading to label words like "the" as entities leading to false positives. Recall measures how many of the coreferences in the gold standard were correctly labelled. This too,performs poorly with a recall of 44.8%. Many of the references, especially over long spans, were missed, given that some of my articles were quite long. The Model was also worse at tokenization than spacy and struggled with entities longer than two words, such as with full names and therefore also incorrectly corefencing them.

3c) Refer to **ner_analyse.ipynb**
The dataset included **9,741 named entities**
The following plot highlights the distribution of the Entity Types.

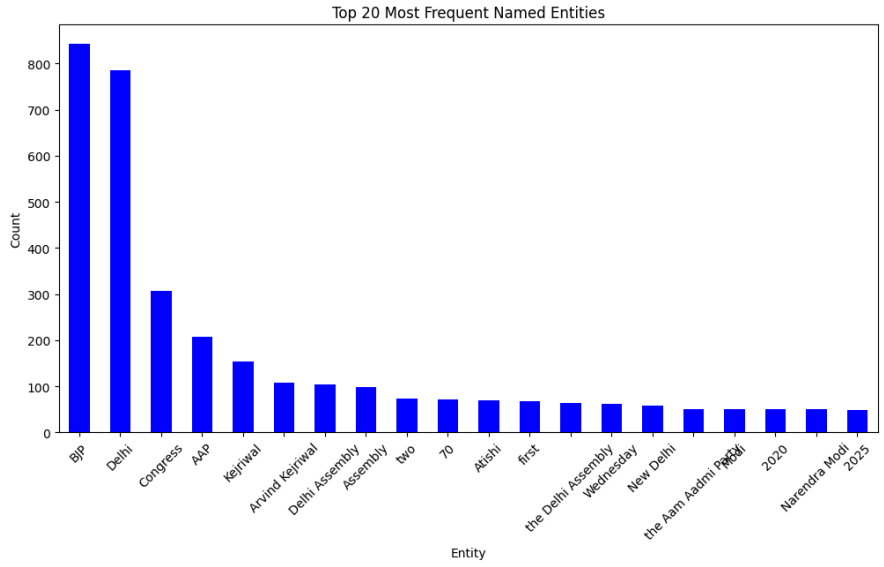


Organisation is the most common entity in the dataset, with almost a third of the entities representing them. This is largely because the party names especially, the BJP and AAP appeared frequently in each article. Organisation is even more common than name, suggesting that the election or at least the coverage of it was heavily based on party narratives rather than candidates.. Person is the second most common entity largely from the candidates contesting the poll, as well as interviews of voters. Geopolitical entity(GPE) appears next due to the frequent mention of Delhi and the constituencies where the elections were conducted. The high number of dates was to be expected given these were newspaper reports.

Of all the entities, these were the most frequent:

Entity	
BJP	842
Delhi	785
Congress	307
AAP	207
Kejriwal	154
Arvind Kejriwal	108
Delhi Assembly	104

The BJP is by far the most frequently mentioned entity, getting more than 4 times the number of mentions as AAP. This suggests that much of the coverage was dominated by the BJP, which won the election.



There were also significantly high mentions of Arvind Kejriwal and Atishi compared to any BJP candidates, suggesting that the election coverage was more party focused than candidate focused for the BJP.

However there are many insights which NER itself cannot provide such as

1. **Sentiment Analysis** on entity mentions. For example whether coverage on BJP was positive or negative.

2. **Coreferences -** Different mentions of the same entity need to be resolved, which can lead to incorrect conclusions such as AAP and Aam Aadmi Party.

3. **Entity Relationships -** NER alone cannot tell us how different entities or their mentions relate to each other. Such as if Congress and BJP were mentioned together.
4. **Topic Modelling**- NER cannot by itself tell you which topic the Entity were mentioned in the context of.