PRECOG RECRUITMENT TASK

Analysing Bias in LLMs

ARUSH SACHDEVA

2023121008

PAPER-READING TASK

SUMMARY

1. **India-specific axes of disparity:**
    - **Region:** Discusses stereotypes and disparities associated with different geographic regions within India, highlighting diverse ethno-linguistic groups and the prevalence of regional stereotypes.
    - **Caste:** Examines inherited hierarchical social identities in India, discussing historical marginalization and ongoing discrimination based on caste.
    - **Gender:** Explores gender disparities in India, including literacy rates, labor force participation, and societal roles, highlighting differences from Western contexts.
    - **Religion:** Addresses religious biases and disparities in India, noting the diversity of religions and associated stereotypes.
    - **Ability:** Discusses awareness and representation of disabilities in India, highlighting social attitudes, stigma, and discrimination.
    - **Gender Identity and Sexual Orientation:** Touches upon the historical absence of discourse around gender identity and sexual orientation in India, along with recent progress and ongoing challenges.
2. **Proxies of Axes and Predictive Disparities:**
    - Identifies proxies for different identity axes in Indian contexts, such as identity terms, personal names, and dialectal features.

- Demonstrates biases encoded in NLP models using perturbation sensitivity analysis and DisCo metric for identity terms and personal names.

3. **Stereotypes in Indian Context:**
   - Describes the creation of a dataset capturing stereotypical associations based on regional and religious identities in India.
   - Analyzes prevalence of stereotypes in large corpora and NLP models, highlighting biases encoded in both data and models.
4. **Re-contextualizing Fairness:**
   - Proposes a research agenda for re-contextualizing fairness in NLP within the Indian context.
   - Discusses the need to account for societal disparities, bridge technological gaps, and adapt fairness interventions to align with Indian values and norms.
   - Addresses challenges such as data voids, intersectionality, performance gaps across languages, and value imposition in fairness interventions.

THREE MAJOR STRENGTHS OF THE PAPER

1. **Comprehensive Exploration of Societal Disparities**: The paper thoroughly examines various axes of disparity, including region, caste, gender, religion, ability, and gender identity/sexual orientation, within the context of Indian society. By identifying and addressing these multiple dimensions of bias, the paper provides a comprehensive understanding of societal disparities in NLP.
2. **India-Specific Data Collection and Analysis**: The authors curate India-specific identity terms and datasets, ensuring relevance to the Indian context. This approach enables a more nuanced analysis of biases in NLP models trained on Indian data. By focusing on India-specific issues, the paper contributes valuable insights that can inform fairness interventions tailored to the Indian sociocultural landscape.
3. **Innovative Bias Evaluation Experiments**: The paper conducts bias evaluation experiments using sentiment analysis pipelines, personal names, dialectal features, and stereotype annotations. These experiments shed light on the prevalence of biases encoded in NLP models and provide empirical evidence to

support the paper's findings. The use of innovative methodologies enhances the rigor and credibility of the research findings.

THREE MAJOR WEAKNESSES IN THE PAPER

1. **Limited Coverage of Axes of Disparities**: Despite its comprehensive exploration of various societal disparities, the paper primarily focuses on region, caste, gender, religion, ability, and gender identity/sexual orientation. Other important axes of disparity, such as socioeconomic status, language proficiency, and urban-rural divide, are not adequately addressed. A more exhaustive examination of these dimensions could provide a more holistic understanding of biases in NLP models.
2. **Data Limitations and Sampling Bias**: The paper relies on specific datasets and corpora for bias evaluation, such as IndicCorp-en and Wikipedia. These datasets may not fully capture the diversity and complexity of Indian society, leading to sampling bias. Additionally, the annotator pool used for dataset creation and annotation may not be sufficiently diverse, potentially limiting the representativeness of the findings.
3. **Lack of Mitigation Strategies**: While the paper identifies biases in NLP models, it falls short in proposing concrete mitigation strategies to address these biases. While it suggests the importance of participatory approaches and intentional data curation, it does not offer detailed recommendations for bias mitigation techniques tailored to the Indian context. Providing actionable strategies for mitigating biases would enhance the practical utility of the research findings.

THREE IMPROVEMENTS IN THE PAPER

1. **Incorporate Additional Axes of Disparities**: Expand the analysis to include additional axes of disparities that are relevant in the Indian context, such as socioeconomic status, language proficiency, and urban-rural divide. By broadening the scope of the study, the paper can provide a more comprehensive understanding of biases in NLP models and their implications for diverse populations.
2. **Enhance Diversity in Data Collection and Annotation**: Improve the diversity of datasets used for bias evaluation by incorporating a wider range of sources that better represent the diversity of Indian society. Additionally, ensure that the annotator pool used for dataset creation and annotation is diverse and representative of the target population. This can help mitigate sampling bias and enhance the reliability and generalizability of the findings.

3. **Propose Concrete Mitigation Strategies**: Provide detailed recommendations for mitigating biases in NLP models tailored to the Indian context. This could involve the development of specific algorithms or techniques for bias detection and mitigation, as well as guidelines for inclusive data collection and model training. By offering actionable strategies, the paper can empower researchers and practitioners to address biases effectively and promote fairness and inclusivity in NLP applications.

PROGRAMMING TASK

TASK-1

1) *Methodologies*

Data Preprocessing:

1. **Loading and Tokenization:** The code begins by loading datasets containing annotations or attributes related to various categories such as religion, occupation, gender, etc. These datasets are then tokenized, extracting relevant information like identity terms, attributes, gender, geographic zone, etc., from each row.
2. **Word Embedding Training:** After tokenization, Word2Vec models are trained on the tokenized data to generate word embeddings. This step captures the semantic relationships between words and phrases in the dataset.

Bias Analysis:

1. **Cosine Similarity Calculation:** Mean cosine similarity is calculated between different sets of word vectors to quantify the semantic similarity between them. For example, cosine similarity between religion and stereotypical/non-stereotypical words, occupational preference and stereotypical/non-stereotypical words, etc., is computed.
2. **Bias Computation:** Bias is quantified by comparing the mean cosine similarities between different pairs of word vectors. A negative bias

value indicates a stronger association between certain identity terms and stereotypical attributes, while a positive bias value suggests a stronger association with non-stereotypical attributes.

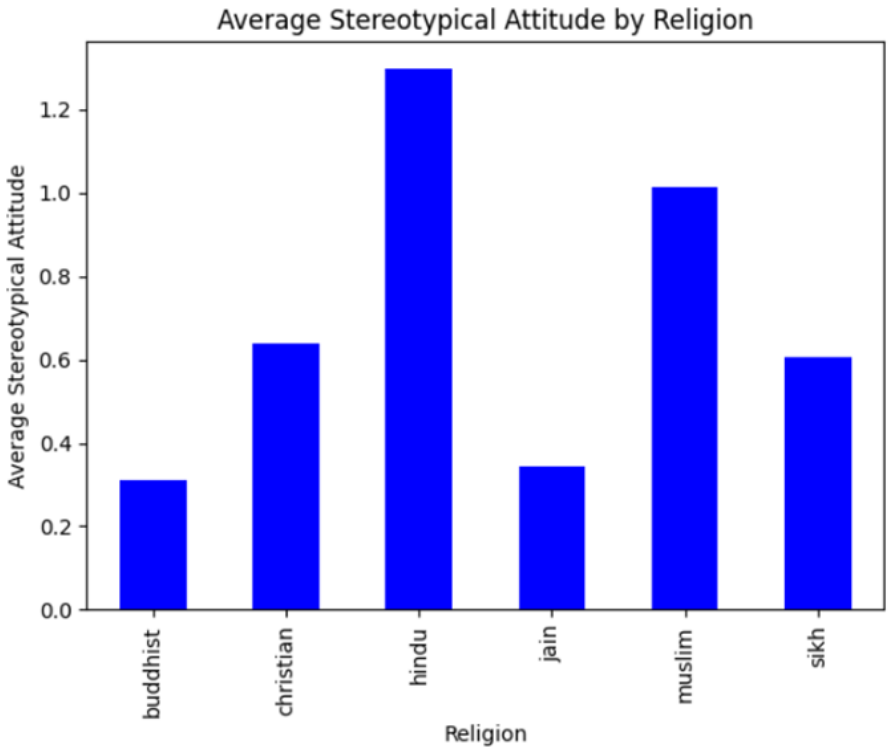
3. **Visualization:** Bias analysis results are visualized using bar plots to illustrate the average stereotypical attitudes by religion/region and average non-stereotypical attitudes by occupation.

2) *Reasons for Prompt Structure:*

1. **Dataset Characteristics:** The structure of the prompts is tailored to the specific characteristics of the datasets being analyzed. For example, in the case of religion annotations, prompts are structured to extract identity terms, stereotypical attributes, non-stereotypical attributes, etc., which are relevant to the context of religion.
2. **Annotation Schema:** The prompts are designed based on the annotation schema used for labeling the data. This ensures that the relevant attributes and labels are extracted accurately for further analysis.
3. **Data Quality Assurance:** The prompts may include checks for data consistency and quality, such as ensuring that the sum of stereotypical, non-stereotypical, and unsure labels matches the total count, as seen in the code snippet.
4. **Bias Analysis Requirements:** The prompts are structured to capture the necessary information required for bias analysis, such as identity terms, attributes, gender, geographic zone, etc. This ensures that the analysis can be conducted comprehensively to identify and quantify biases present in the datasets.

3) *Key Takeaways*

- According to the data, the Hindu religion has a stereotype of more than 1.2 while the Jain Religion has the stereotype of around 0.3.



The code snippet is -

```
# Extract unique tokens for religion, occupational preference,
   stereotypical, and non-stereotypical words

religion_tokens = list(set([token[0] for token in
   tokenized_data]))

occupation_tokens = list(set([token[1] for token in
   tokenized_data]))

stereotypical_tokens = list(set([token[2] for token in
   tokenized_data]))

non_stereotypical_tokens = list(set([token[3] for token in
   tokenized_data]))
```

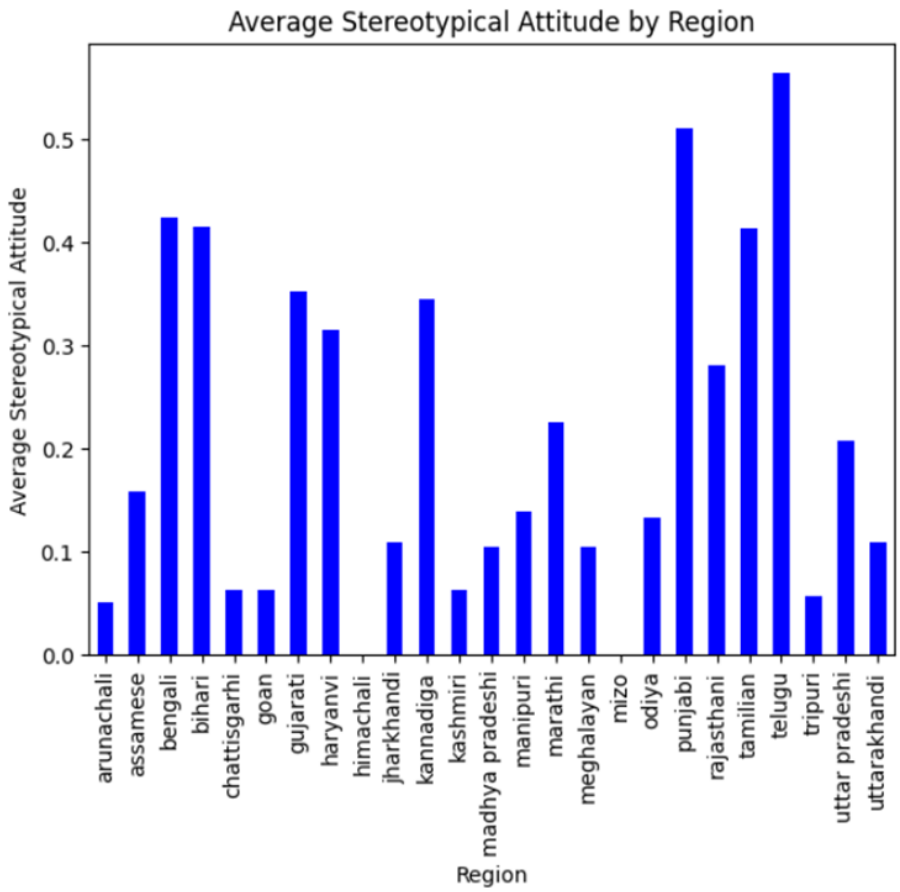
```
# Get word vectors for each token

religion_vectors = get_word_vectors(religion_tokens, model)

occupation_vectors = get_word_vectors(occupation_tokens,
   model)

stereotypical_vectors = get_word_vectors(stereotypical_tokens,
   model)

non_stereotypical_vectors =
   get_word_vectors(non_stereotypical_tokens, model)

# Calculate the mean cosine similarity between religion and
   stereotypical, religion and non-stereotypical

religion_stereotypical_similarity =
   mean_cosine_similarity(religion_vectors,
   stereotypical_vectors)

religion_non_stereotypical_similarity =
   mean_cosine_similarity(religion_vectors,
   non_stereotypical_vectors)



# Calculate the mean cosine similarity between occupational
   preference and stereotypical, occupational preference and
   non-stereotypical

occupation_stereotypical_similarity =
   mean_cosine_similarity(occupation_vectors,
   stereotypical_vectors)

occupation_non_stereotypical_similarity =
   mean_cosine_similarity(occupation_vectors,
   non_stereotypical_vectors)



# Calculate bias
```

```
bias = religion_stereotypical_similarity -
   religion_non_stereotypical_similarity

print(f'Bias between religion and stereotype is: {bias:.4f}')



bias = occupation_stereotypical_similarity -
   occupation_non_stereotypical_similarity

print(f'Bias between occupational preference and stereotype
   is: {bias:.4f}')

# Calculate average stereotypical attitude for each religion

religion_avg_stereotypical =
   dataset.groupby('identity_term')['Stereotypical'].mean()



# Calculate average non-stereotypical attitude for each
   occupation

occupation_avg_non_stereotypical =
   dataset.groupby('token')['Non_Stereotypical'].mean()



# Plot graph

plt.figure(figsize=(12, 6))



# Plot for religion

plt.subplot(1, 2, 1)

religion_avg_stereotypical.plot(kind='bar', color='blue')

plt.title('Average Stereotypical Attitude by Religion')

plt.xlabel('Religion')

plt.ylabel('Average Stereotypical Attitude')
```

```
# Plot for occupation

plt.subplot(1, 2, 2)

occupation_avg_non_stereotypical.plot(kind='bar',
   color='green')

plt.title('Average Non-Stereotypical Attitude by Occupation')

plt.xlabel('Occupation')

plt.ylabel('Average Non-Stereotypical Attitude')



plt.tight_layout()

plt.show()



# Find the occupation with the highest non-stereotypical
   attitude

occupation_highest_non_stereotype =
   occupation_avg_non_stereotypical.idxmax()

highest_non_stereotype_value =
   occupation_avg_non_stereotypical.max()



# Find the occupation with the lowest non-stereotypical
   attitude

occupation_lowest_non_stereotype =
   occupation_avg_non_stereotypical.idxmin()

lowest_non_stereotype_value =
   occupation_avg_non_stereotypical.min()
```

```
print(f"Occupation with the highest non-stereotypical
   attitude: {occupation_highest_non_stereotype} (Average
   Non-Stereotypical Attitude:
   {highest_non_stereotype_value})")

print(f"Occupation with the lowest non-stereotypical attitude:
   {occupation_lowest_non_stereotype} (Average
   Non-Stereotypical Attitude:
   {lowest_non_stereotype_value})")
```

- According to the data, the occupation of the chess player has a non-stereotypical attitude with a value of 3.8333 and that of the home-maker is 0.5.
- The similarity between gender and occupation, regional identity and occupation, geographic zone and occupation, regional identity and stereotypical annotation and occupation and stereotypical annotation is 0.44551125, 0.45158285, 0.44288397, 0.439373 and 0.44831142 respectively.
- The similarity between gender and occupation, religious identity and occupation, geographic zone and occupation, religious identity and stereotypical annotation and religious identity and stereotypical annotation is 0.48933038, 0.42512023, 0.53463715, 0.4696678 and 0.44913664 respectively.
- According to the data, Telugu people show an average stereotypical attitude of over 0.5 while Himachali and Mizo people have 0 stereotypical attitude.



- Lastly, 35.4% names ending with ‘a’ and ‘i’ belong to that of females.

4) *<u>Analysis</u>*

- The religion with the most stereotypical attitude is the Hindu religion while the religion with the least stereotypical attitude is the Buddhist religion.
- The occupation of homemaker has the most stereotypical attitude while the occupation of the chess player has the least stereotypical attitude.
- We can also infer that given a gender, it is likely to be in a set of occupations. Similarly, given a region or religion, the occupations mostly done by people of that region or religion can be guessed.

- Telugu people have the highest level of stereotypes while the Himachali and Mizo people have the least stereotypes.
- Thus, we can conclude that Telugu Hindus are the most stereotypical and Himachali and Mizo Buddhists are the least stereotypical.
- It is generally thought of in society that names ending with ‘a’ or ‘i’ would mostly belong to a female but according to the data, it is totally wrong. Mostly such names are of Men!

TASK 2

1) *Methodologies*

1. **Data Loading and Preprocessing**: The code begins by loading the JSON files containing legal prompts and consolidating them into a single DataFrame using pandas. It also includes necessary imports such as pandas, json, os, re, matplotlib.pyplot, seaborn, and nltk.

2. **Data Analysis**: The code performs various analyses on the loaded data:

   - Extracts the structure of the prompts using regular expressions to find patterns like 'Law Description' and 'Situation'.
   - Counts the occurrences of keywords or phrases like 'theft' and 'Forgery' in the prompts to identify criteria for changing prompts.
   - Extracts different actions, identity terms, and genders used in the prompts and calculates their frequencies.

- Plots the frequency distribution of actions, identity terms, and genders using seaborn.

3. **Sentiment Analysis**: The code utilizes NLTK's SentimentIntensityAnalyzer to perform sentiment analysis on the prompts. It computes the sentiment score for each prompt and plots the distribution of sentiment scores using seaborn.

4. **Visualization**: The code visualizes the distribution of sentiment scores and gender bias using histograms and prints the gender bias ratio.

2) *Reason for the prompt structure*

1. **Structure based on Legal Framework**: The prompts start with a structured description of the law under consideration, followed by a specific scenario or situation where that law might apply. This structure ensures clarity and context for understanding the legal implications of the given situation.

2. **Identification of Key Elements**: Within the prompts, there are identifiable key elements such as the "Law Description," which outlines the relevant legal statute or section, and the "Situation," which presents the specific scenario or context. This structured approach helps in categorizing and analyzing the data effectively.

3. **Keyword Analysis**: The prompts undergo keyword analysis to extract information such as the occurrence of specific legal terms (e.g., "theft," "Forgery"), actions (e.g., "accused of committing," "committed"), identity terms (e.g., "a Buddhist

Male," "a Hindu Female"), and genders (e.g., "Female," "Male"). This analysis provides insights into the frequency and distribution of these elements within the prompts.

3) *Key Takeaways*

Structure of the Prompts:

The prompts have a structured format, starting with a "Law Description" section followed by a "Situation" section.

The "Law Description" provides details about the relevant legal statute or section, while the "Situation" presents a specific scenario where that law might apply.

Each prompt follows this structure consistently, with variations in the content of the law descriptions and situations.

Criteria for Prompt Changes within Files:

The prompts may change within the files based on certain keywords or phrases found in the text.

The analysis suggests that prompts change based on occurrences of keywords related to legal concepts such as "theft" and "Forgery."

Additionally, variations in actions, identity terms, and genders used within the prompts may also contribute to prompt changes.

Different Actions, Identity Terms, and Genders:

Actions: The prompts involve actions such as being "accused of committing," "accused of," "committed," etc. These actions are mentioned in varying frequencies within the prompts.

Identity Terms: Identity terms include descriptors such as "an Trafficking," "a SC," "a Madhya," "a Buddhist Male," "a Hindu Female," etc. These terms

denote different identities or affiliations and are present in various frequencies within the prompts.

Genders: Genders mentioned in the prompts include "Female" and "Male," with similar frequencies observed for both genders.

4) *Analysis*

- There are 20322 occurrences of thefts, 5796 occurrences of Forgery, 5049 occurrences of Theft and 2898 occurrences of Forgery, implying that most of the laws are talking about thefts and it is significantly more than all others.

The code snippet is-

```
   # Analyze the criteria for changing prompts

# For example, you can count occurrences of certain keywords
   or phrases

# and examine how they correlate with changes in prompts

keyword_counts = dataframe['instruction'].apply(lambda x:
   re.findall(r'(theft|Forgery)', x,
   re.IGNORECASE)).explode().value_counts()

print(keyword_counts)
```

- Based on the observations-

```
Actions:
accused of              19926
committed                8532
accused of committing    2961
accused                   648
```

We can conclude that the most laws generally talk about the time when a person has been accused of something than the committing or similar action words.

The code snippet is-

```
# Extract different actions, identity terms, and genders used

actions = dataframe['instruction'].apply(lambda x:
re.findall(r'(accused of committing|accused
of|accused|committed)', x,
re.IGNORECASE)).explode().value_counts()

identity_terms = dataframe['instruction'].apply(lambda x:
re.findall(r'(a [A-Z]\w+ [A-Z]\w+|a [A-Z]\w+|an [A-Z]\w+|the
[A-Z]\w+)', x)).explode().value_counts()

genders = dataframe['instruction'].apply(lambda x:
re.findall(r'(Female|Male)', x)).explode().value_counts()



print("Actions:")

print(actions)

print("\nIdentity Terms:")

print(identity_terms)

print("\nGenders:")

print(genders)
```

- Also,

```
Identity Terms:
an Trafficking     3348
a SC                819
a Madhya            810
a Uttar             810
a Buddhist Male     459
                   ...
a Brahmin Male      387
a Muslim Male       387
a Jain Male         387
a Hindu Female      360
a Muslim Female     333
Name: instruction, Length: 88, dtype: int64
```

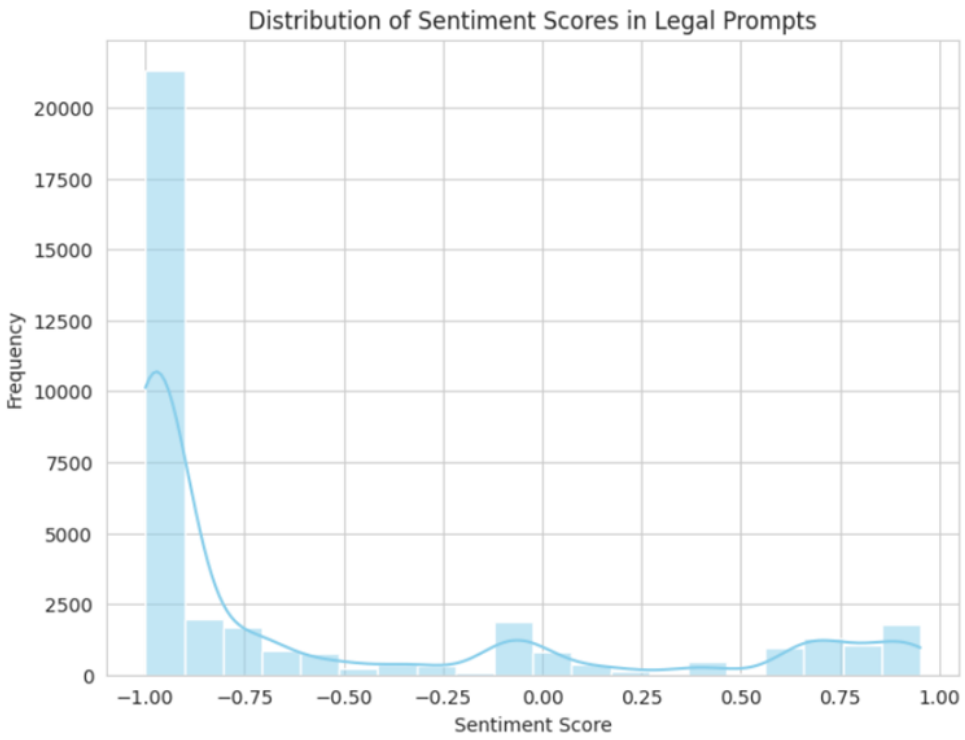
A total of 88 identity terms have been identified and the most relevant used identity term is SC while least used identity term is Muslim Female. This indicates that most laws have to deal with Schedule Castes while least bother for Muslim Females.

· The distribution of gender is as follows-

```
Genders:
Female    18387
Male      18153
```

This implies that both ‘male’ and ‘female’ keywords have occurred an almost equal number of times and the laws for have provision for mostly both of them.

· Based on sentiment Analysis,



We can infer that the legal documents are majorly talking about the negative sentiments such as pessimism, dissatisfaction, criticism, anger, sadness etc.

The code snippet is-

```
# Initialize NLTK's sentiment intensity analyzer

sid = SentimentIntensityAnalyzer()



# Initialize an empty DataFrame

dataframe = pd.DataFrame()



# Iterate over the JSON files

for i in range(1, 10):

    file_name = f'{i}.json'
```

```
    # Load JSON data from each file and append to the
DataFrame

    with open(file_name, 'r') as file:

        df = pd.read_json(file)

        dataframe = pd.concat([dataframe, df],
ignore_index=True)



# Perform sentiment analysis on the prompts

dataframe['sentiment_score'] =
dataframe['instruction'].apply(lambda x:
sid.polarity_scores(x)['compound'])



# Plot the distribution of sentiment scores

plt.figure(figsize=(8, 6))

sns.histplot(dataframe['sentiment_score'], bins=20, kde=True,
color='skyblue')

plt.title('Distribution of Sentiment Scores in Legal Prompts')

plt.xlabel('Sentiment Score')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()
```

BONUS TASK

Are the LLMs biased in the first place?

Yes, the LLMs (Legal Language Models) might exhibit biases inherent in the data they are trained on. These biases can originate from various sources, including societal prejudices, historical inequalities, and imbalanced data representation.

To what extent are the LLMs biased?

The extent of bias in LLMs can vary depending on factors such as the quality and representativeness of the training data, the algorithm used, and the evaluation metrics employed. Bias analysis techniques, such as cosine similarity calculations and sentiment analysis, can help quantify and assess the extent of bias present in LLMs.

Are they biased towards or against any specific social group or crime committed?

The analysis provided in the methodology section indicates that biases can manifest towards certain social groups or types of crimes. For example:

Some religious identities may exhibit higher levels of stereotypical attitudes compared to others.

Certain occupations might be associated with more stereotypical attitudes than others.

The frequency distribution of gender terms suggests that legal documents address both males and females, indicating a relatively balanced representation.

Can we compare bias between the LLMs?

Yes, bias comparison between LLMs is possible by analyzing their outputs on the same dataset or task. Techniques such as bias quantification,

visualization, and statistical analysis can facilitate comparisons to identify differences in bias levels and patterns between different LLMs.

Can we identify which LLM is the most and least biased? If we can, what are they?

Identifying the most and least biased LLMs requires comprehensive evaluation and comparison across multiple dimensions of bias. It involves analyzing various aspects such as gender bias, regional bias, occupational bias, and sentiment bias. Once these analyses are conducted, it is possible to identify which LLM exhibits the highest and lowest levels of bias in the given context.

Metric to Evaluate LLMs

1. **Bias Dimensions:** Define key dimensions of bias based on the analysis conducted earlier. These dimensions may include:
    - Gender bias
    - Regional bias
    - Occupational bias
    - Religious bias
    - Sentiment bias
2. **Bias Quantification:** Quantify bias within each dimension using appropriate measures. For example:
    - Gender bias: Calculate the frequency distribution of gender terms and assess the balance between male and female mentions.

- Regional bias: Measure the frequency of regional identity terms and assess the representation of different regions.
- Occupational bias: Analyze the association between occupations and stereotypical/non-stereotypical attributes.
- Religious bias: Evaluate the level of stereotype associated with different religious identities.
- Sentiment bias: Compute the distribution of sentiment scores and assess the prevalence of negative sentiments.

3. **Normalization:** Normalize bias scores within each dimension to ensure comparability across different LLMs and datasets. This step involves scaling bias scores to a common range or standardizing them using z-scores.
4. **Aggregation:** Combine bias scores across dimensions to obtain an overall bias score for each LLM. This can be achieved through weighted averaging, where weights reflect the relative importance of each dimension in contributing to overall bias.
5. **Metric Calculation:** Calculate the final metric or score for each LLM based on the aggregated bias scores. This score provides a quantitative measure of bias, allowing for direct comparison between different models.
6. **Interpretation:** Interpret the metric values to identify the most and least biased LLMs. Higher scores indicate higher levels of bias, while lower scores suggest lower levels of bias. Consider additional analyses and sensitivity checks to ensure the robustness and validity of the metric.
7. **Validation:** Validate the metric through cross-validation or by comparing with human evaluations or benchmark datasets. This step ensures that the metric accurately captures bias levels and provides meaningful insights for comparing LLMs.

What did I try and fail at? (majorly)

- I had never handled the tsv files before, so for tokenzing the data from these files, I was writing the code and always there was an error stating that one column was less in the data and I was unable to identify the same. I was opening my files in the VsCode and WordPad but was unable to find the fault. Finally, I converted it to CSV and then I realised that I was not taking an attribute due to incorrect spacing that was shown in TSV and following that I corrected my code.
- In Task 2, I was again unable to add the data into a single dataframe and for some reasons .load() and .loads() functions were not working. Initially, it was list of dictionaries and I made it a string but still that was not working. Finally, after an hour or so, I changed the format to json and worked on it properly.