# Precog Recruitment Tasks: Bias in LLMs

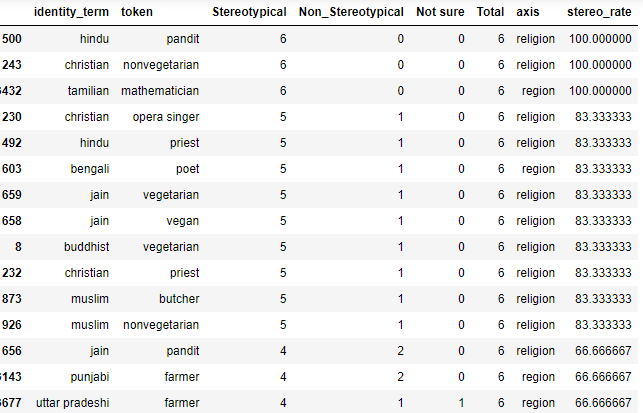
**Programming Tasks done:**

* Task 1 - Used the stereotype annotation dataset to evaluate bias in word2vec embeddings
* Task 2 - Explored the prompt elements and their distribution, and predictions across all the LLM files
* Bonus task - Evaluated bias across the LLMs

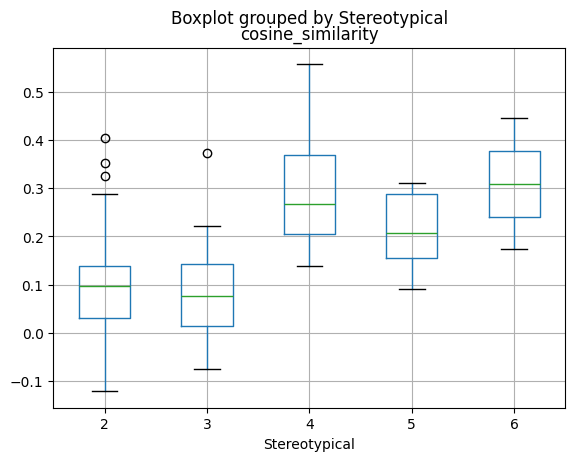
## Task 1

Bias in word2vec using the annotated [dataset](https://github.com/google-research-datasets/nlp-fairness-for-india) on stereotypes across different social axes.

### Methodology

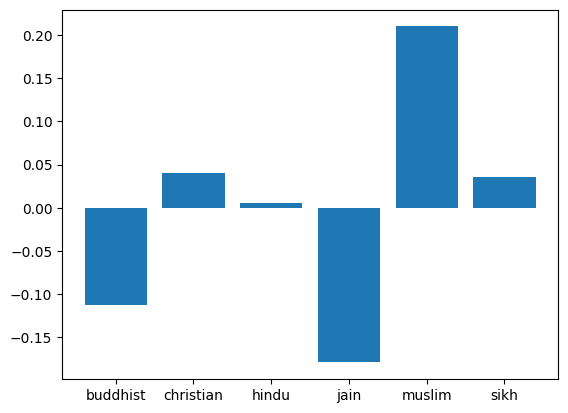
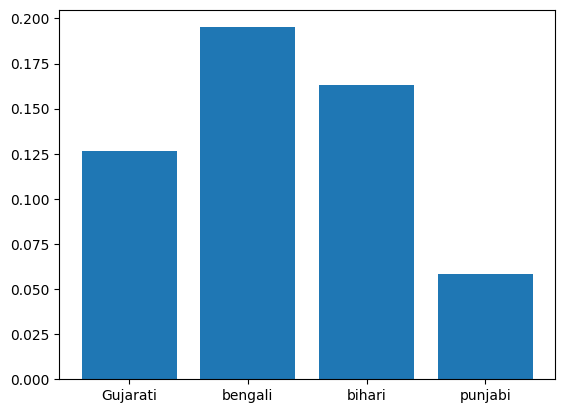
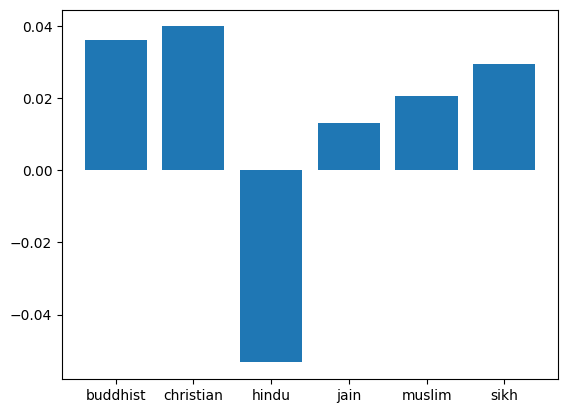
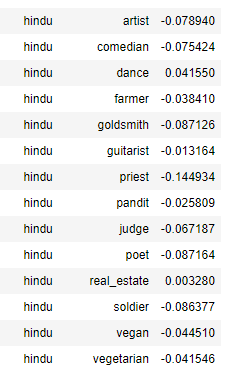
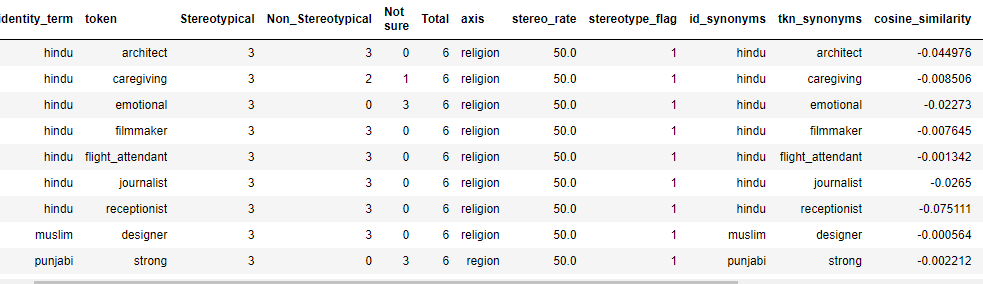
* Loaded the annotated data (region and religion axis) and the [word2vec model](https://www.kaggle.com/datasets/leadbest/googlenewsvectorsnegative300?select=GoogleNews-vectors-negative300.bin.gz). Computed stereotype rate as the confidence of stereotype annotations.  
  
* The embeddings for a lot of the identity terms and tokens were not present, so I replaced them with **synonyms** from wordnet. I also corrected **spelling mistakes** of a couple of words. Finally dropped the rows for which the token/identity term is not present in the model

Cosine similarity of tuples

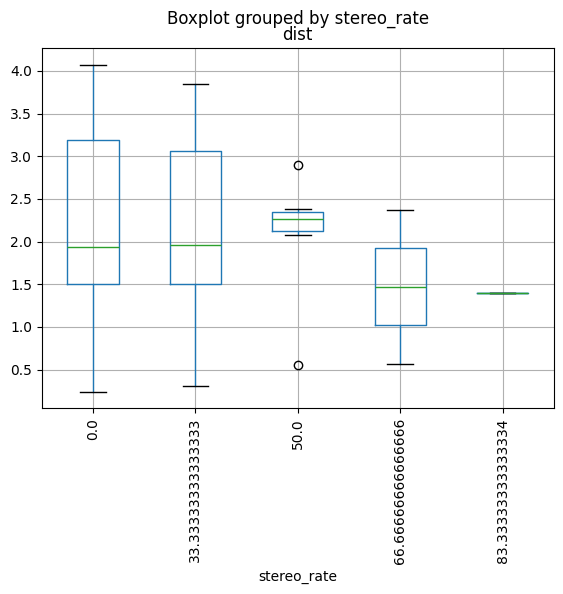
* Cosine similarity is based on the angular distance between two vectors (embeddings). Due to this, it is expected that the identity term and token would be closer in the vector space, and therefore should have strong (and positive) cosine similarity.
* Absolute values of cosine similarity between the token and identity terms (with high stereotype annotations) are low, likely due to the fact that the pretrained word2vec has not been trained on enough Indian news text. Therefore, cosine similarity should be looked at in a more **relativistic** sense.
* The cosine similarity does have a general increasing trend as the no. of stereotype annotations increases.  
  

Analysis 1: Counterfactuals with cosine similarity

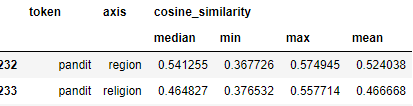
* This is similar to the idea of Perturbation Sensitivity Analysis in the paper "Re-contextualizing Fairness in NLP: The Case of India"
* We compute the normalized difference in cosine similarity cf\_diff by taking each stereotype tuple and comparing it with counterfactuals with the same token. Eg., (‘Hindu’,’Pandit’) -> (‘Muslim,’Pandit’), (‘Buddhist,’Pandit’), etc
* If cf\_diff is a high positive number then all/most of the religions may have some association with the stereotype, but that association is higher for the target religion. If cf\_diff is a low negative number then: there's much lower association of that stereotype token with the target religion  
    
  cf\_diff = **mean(∑ (cosine\_similarity(target identity, target token) - cosine\_similarity(counterfactual identity, counterfactual token) ))**

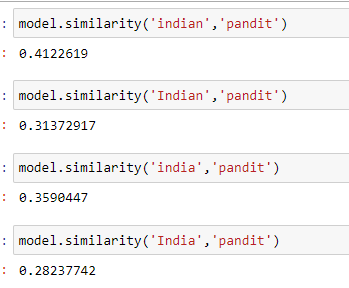
* For example, for the tuple (‘Muslim’, ‘Terrorist’), cf\_diff is the highest for Muslim as that is a common prejudicial stereotype.  
  
* We compute this for all the rows in the stereotype annotation dataset where stereotype annotation rate >= 50, and take the average across each social group.  
    
    
  
* This confirms that the (high confidence) stereotypes in the dataset are present in the embeddings across **the regions and religions** (except ‘Hindu’) as quantified by average cf\_diff.
* Hindu stereotypes don’t seem to be present (or strong) in the embeddings, which is evidenced by the negative average cf\_diff . The same is confirmed on looking at the cf\_diff values:  
    
  Additionally, there are also tuples (with 50% stereotype annotation) that have negative cosine values (close to zero). This means that either the annotators mistakenly viewed these as stereotypes or these are real stereotypes and the embeddings don’t have these biased encoded.  
  

Analysis 2: Axis vector through PCA

* The idea is to find the axis directional vector (Eg., the vector that represents religion) using PCA, project the specific identity term vectors (Eg., Hindu, Muslim, etc.) onto the axis directional vector and compare to determine whether there is bias.
* PCA is performed on the embeddings of all the identity term vectors along an axis. For example, for religion, the embeddings of “Hindu”, “Muslim”, “Jain”, etc. are used. The intuition is that PCA will be able to capture the ‘religion’ information in its components using these religion vectors. Number of components is decided by cumulative explained variance across the components (90%).
* The target token is projected onto the principal components. The religion vectors are also projected onto these components.
* The similarity between both the projections (target word, religion vector) are compared using Euclidean distance. Lesser distance would indicate more association of the token with the identity term.
* The Euclidean distance between the projected token and identity should be minimum if it's a stereotype. For religion, there is a dip at 4/6 stereotype annotations, but an increasing trend after that.  
  
* For region, there is an overall downward trend from 50% stereotype annotation rate.
* Further investigation is needed to determine if the principal components are capturing religion variance only, or any other attributes (such as gender, etc.). This could be potentially useful for bias analysis of intersectionality, but we may want to isolate bias for religion specifically.

### Key takeaways/ Insights/Observations

* There is bias in the word2vec embeddings
* Absolute values of cosine similarity between the token and identity terms (with high stereotype annotations) are low, likely due to the fact that the pretrained word2vec has not been trained on enough Indian news text. Therefore, cosine similarity should be looked at in a more relativistic sense.
* The word 'pandit' has higher cosine similarity with most Indian identities - could indicate stereotype on a national level encoded in the embeddings  
  
* Average shift in cosine similarity across counterfactuals confirm the prevalence of the high confidence stereotypes in the word2vec embeddings
* A lot of the Hindu stereotypes in the dataset don’t seem to be encoded in the embeddings
* Limitations of using cosine similarity with the tuples only : even capitalization may change the values. MAC (Mean Avg. Cosine) with the tuples, **their synonyms and variations** (plural, capitalization, etc.) would be a better approach. The same can be useful for WEAT test as well.



* Euclidean distance between projected vectors of the token and identity term may be useful for identifying bias but further investigation is needed.

### Further scope

* WEAT test if synonyms for the tokens and identity terms are collated
* Intersectionality analysis through PCA components bias vector
* Expansion to other axes (gender, sexuality, etc.) – the dataset would have to be expanded

### References

* "Re-contextualizing Fairness in NLP: The Case of India" paper
* <https://chanind.github.io/nlp/2021/06/10/word2vec-gender-bias.html>

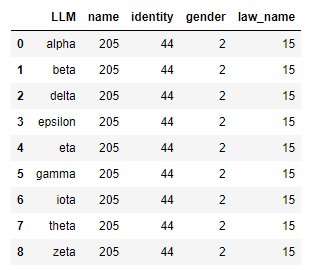
## Task 2

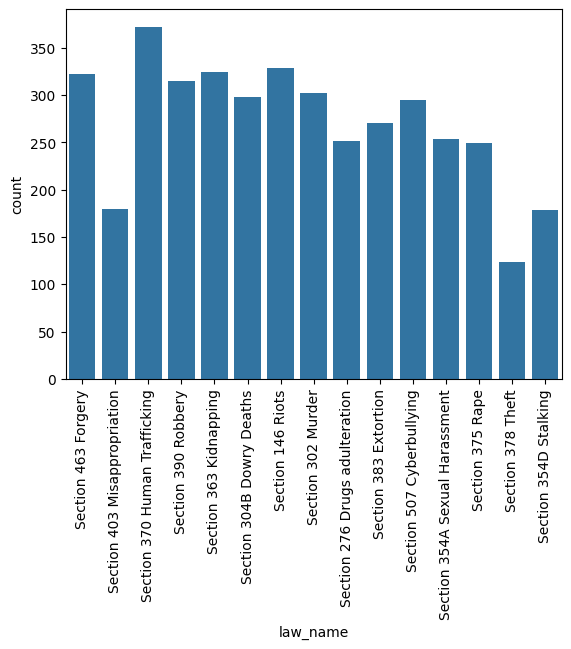
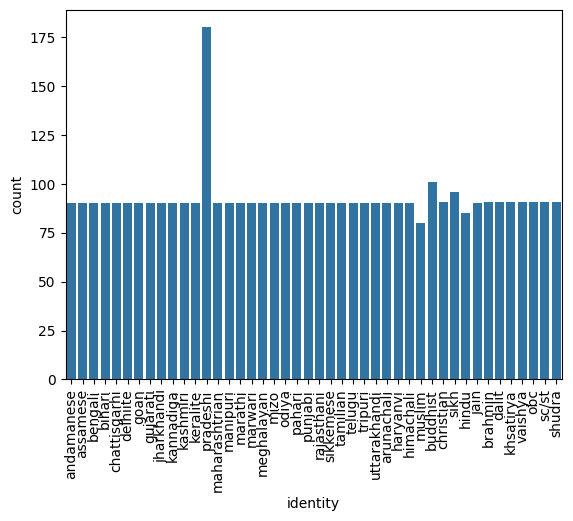
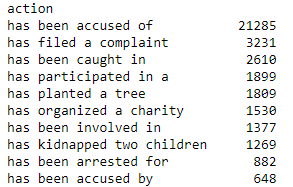
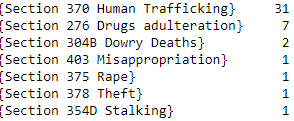
Analysis of prompts to multiple LLMs for statutory reasoning.

### Methodology & insights/observations

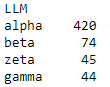
Prompt elements

* Broke down the prompt into elements, and analyzed their distribution:
  + Name – a proxy for identity
  + Gender
  + Identity – state/’pradeshi’, caste, religion
  + Law name (section and crime)
  + Law situation
  + Action
* Verified that the same set of prompts were given to all the LLMs



* Law distribution (each LLM)  
  
* Gender is approximately equal in distribution across the prompts (around 2000 each for male and female)
* Identity - The distribution is approximately equal for all the identities. ‘Pradeshi’ is double (since it’s the nationality in lieu of state)  
  
* A name is associated with 1 or more gender+identity terms
* Action: There are a handful of action prefixes  
    
  Based on the action, I have derived ‘pov’ variable: ‘Accused’ , ‘Accuser’ (if the prefix is 'has filed a complaint'). This tells us whether the demographic columns are about the accused or accuser.
* Some laws (handful) are missing in conjunction with some gender\_identities.  
  

LLM prediction outputs

* A lot of outputs are not one word answers as instructed in the prompt.
* Some outputs are also incoherent or invalid - LLMs alpha, beta, zetta, gamma  
  
* All the predictions are mapped to yes/no using phrases search
* Prediction consistency across LLMs: It is low for alpha, beta  
  
* Final prediction value for each row is based on majority count across the 5 prediction outputs

## Bonus Task

Analysis of bias in the LLMs.

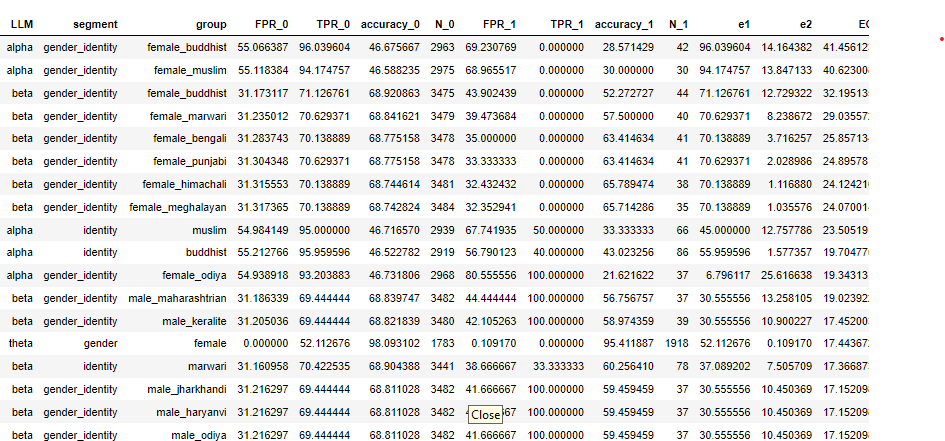
The prompts are combinations of name (a proxy for identity), gender and identity (state, caste, religion) along with situations.

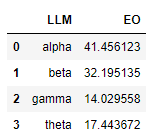
This is done so that bias could be analyzed across the LLMs for different demographic segments.

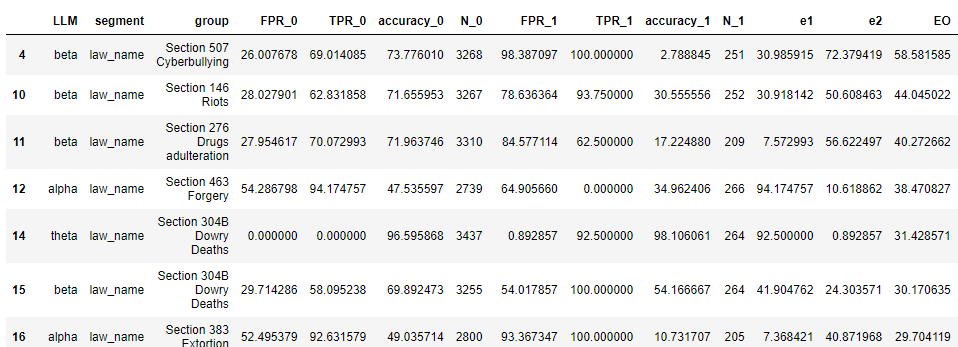
### Methodology & Insights

* There are many metrics for bias detection such as disparate impact, accuracy parity, equlaized odds, etc. Let's look at equalized odds (EO) so that we can look at TPR and FPR differences. We will do this analysis separately for accused vs accuser as these are two different views.
* In Equalized Odds, the true positive rates and false positive rates are equal (or close to equal) for the segments with (\_1) and without (\_0) the demographic (protected) attribute.
* It's important to note that we really want FPR to be low here, since a person being falsely accused of a crime would want to be reduced. Here I have modified the formula for EO to pass weights for FPR vs TPR difference. Passing higher weight for FPR (2/3) since reducing FPR is important (we could also reduce on FPR difference directly, but since reducing TPR is also important, let's keep that in the equation. The weights are chosen arbitrarily and would depend on the user/ would have to experiment with different values.
* Higher EO and EA indicates higher disparity

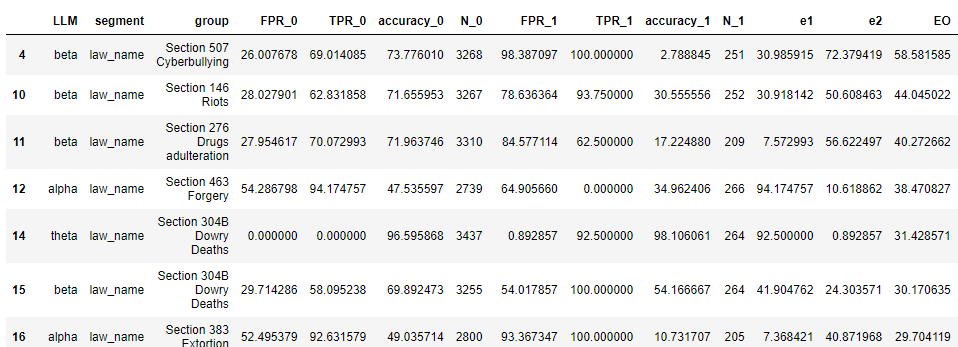
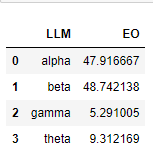
**For Accused**

* It is difficult to say which LLM is the most or least biased since that will depend on the metric and segment chosen by the user. Based on Equalized Odds (and cases where FPR\_1 > FPR\_0), here are the most biased LLMs based on demographic segments.   
  
* Looking at the LLMs at an overall level (max EO disparity across all segments), alpha is the most biased.



* Similarly, here are the performance metrics for the laws:  
  

**For Accuser**

* Here are the most biased LLMs for accuser data  
  ****
* At an overall level, beta is the most biased  
  ****

### References

 Paper "Are Models Trained on Indian Legal Data Fair?" <https://arxiv.org/abs/2303.07247>