Analysis of Code-mixed Sentences

We have a codemixed (with English) dataset from three languages: Hindi, Bengali and Telugu. This data is directly from Facebook, Twitter and Whatsapp, and is thus representative of real world conditions. In this analysis, we normalize the data and look specifically how English is used in each of the three languages.

In [2]:
```
import os
import collections
cwd = os.getcwd()
directory = os.fsencode(cwd)
directory = directory + b'/data'
```

In [3]:
```
data = {"hi":[], "be":[], "te":[]}
```

In [4]:
```
def add_data(ln):
    cur = []
    for sent in data_in_file:
        try:
            word,ln_id,tag = sent.split('\t')
            cur.append([word,ln_id,tag.strip()])
        except:
            if len(cur)==0:
                continue
            data[ln].append(cur)
            cur = []
```

In [5]:
```
for i,filename in enumerate(os.listdir(directory)):
    filename = filename.decode("utf-8")
    file = open(f'data/{filename}')
    data_in_file = file.readlines()
    add_data(filename[3:5])
```

In [6]:
```
results = {"hi":{}, "be":{}, "te":{}}
```

In [7]:
```
for ln in data:
    results[ln]['total_num_sents'] = len(data[ln])
```

In [8]:
```
for ln in data:
    avg = 0
    for sent in data[ln]:
        total = len(sent)
        count = 0
        for word in sent:
            if word[1]=='en':
                count+=1
        avg = (avg+(count/total))
    results[ln]['en_per'] = avg/results[ln]['total_num_sents']
```

In [9]:
```
for ln in results:
    results[ln]['codemixed_factor'] = results[ln]['en_per']/results['be']['er
```

In [10]:
```
pos_tag_counts = {"hi":{}, "be":{}, "te":{}}
```

In [11]:
```
for ln in data:
    for sent in data[ln]:
        for word in sent:
```

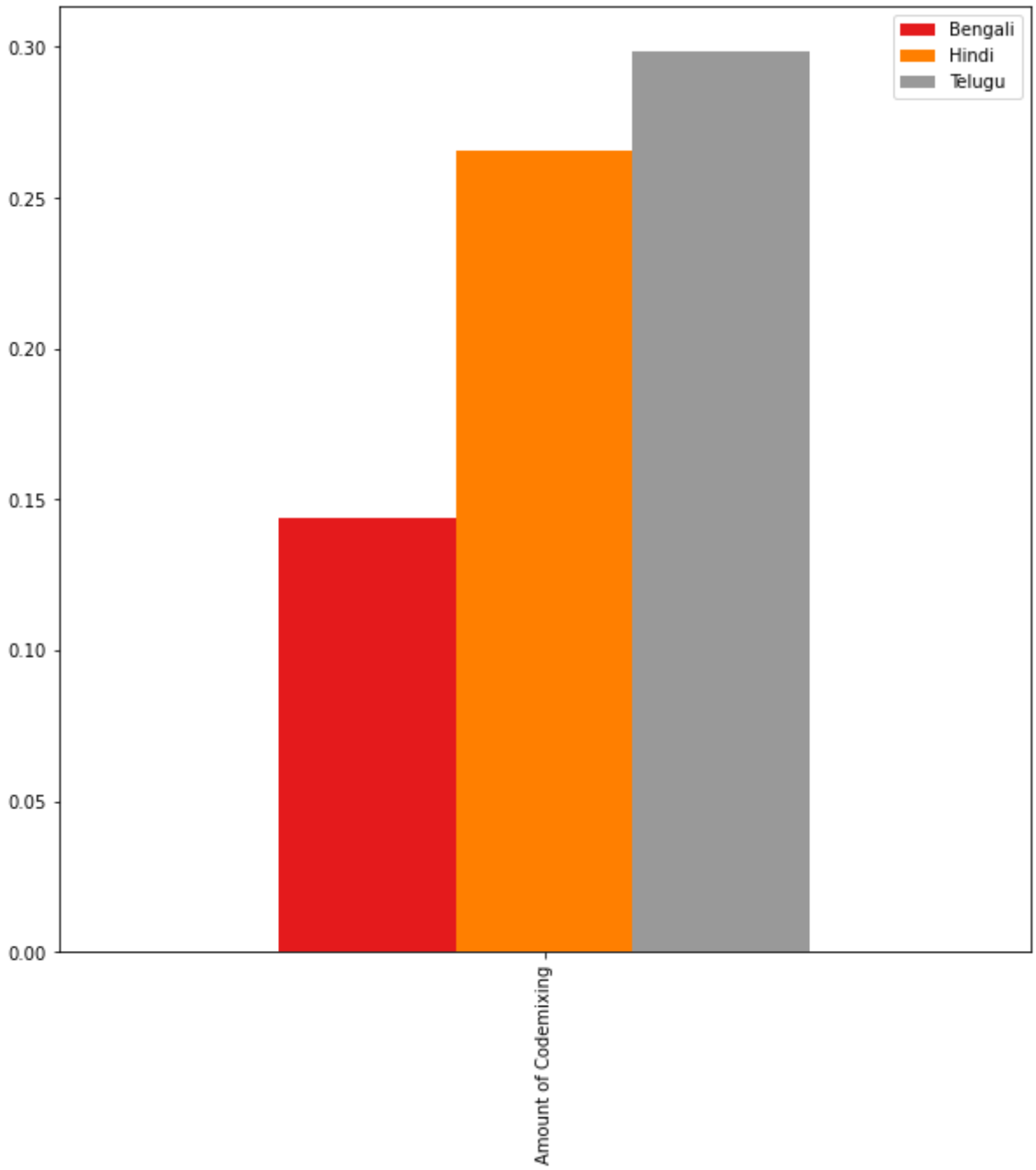
```
            if word[1]=='en':
                if word[2] in pos_tag_counts[ln]:
                    pos_tag_counts[ln][word[2]]+=1
                else:
                    pos_tag_counts[ln][word[2]] = 1
```

We are normalizing the data per two factors:

1. Total Number of sentences
2. Amount of code-mixing

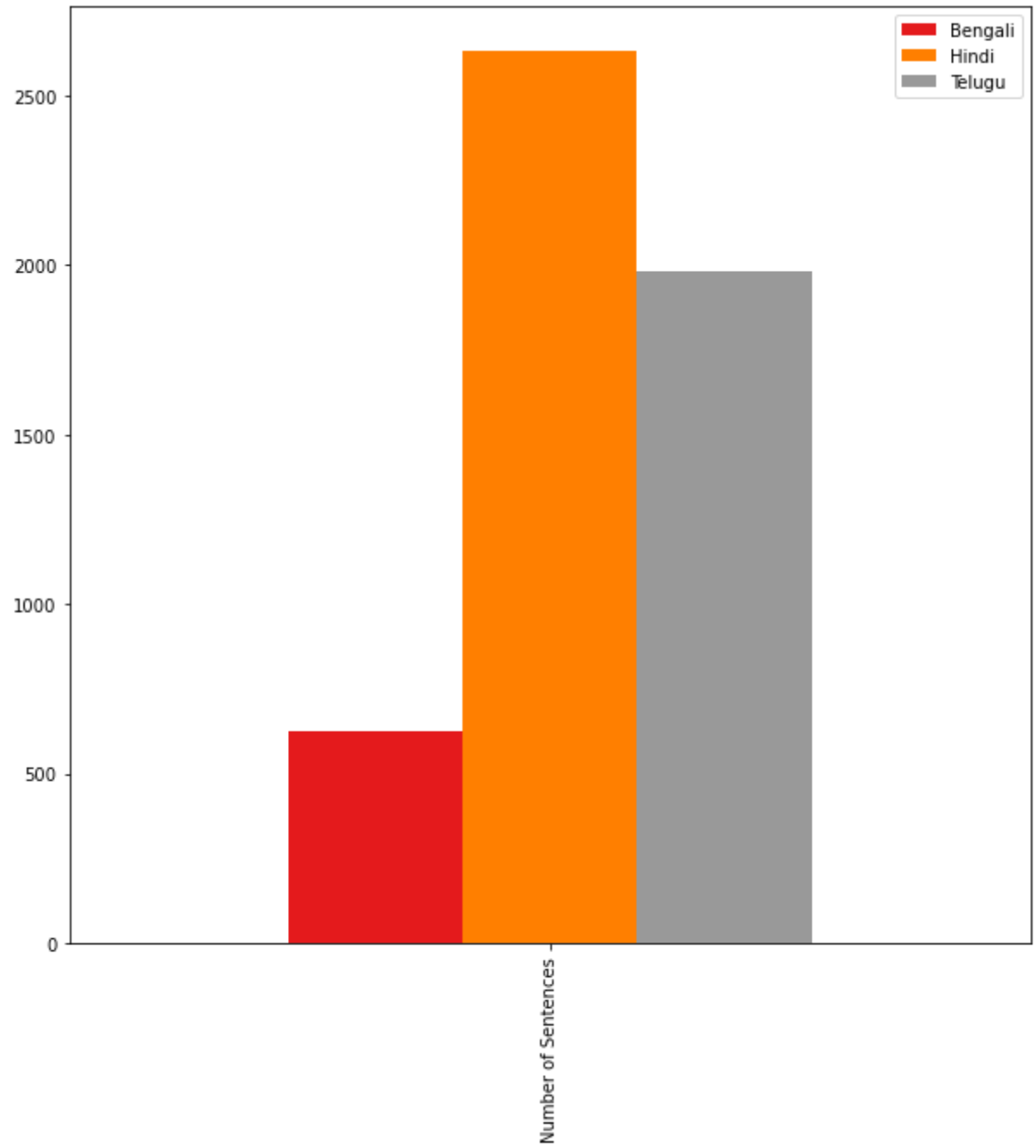
Thus, we look at the normalized counts of how frequent the English POS tags really are.

In [19]:
```
df = pd.DataFrame({"Bengali":results['be']['en_per'],"Hindi":results['hi']['e
```



We can see that Bengali data here has a very low amount of code-mixing (About 14%), while the Telugu data has close to 30% of codemixing. This leads to another interesting analysis, once we normalize the POS tags, the Bengali data is also representative of low code-mixing.

In [18]:
```
df = pd.DataFrame({"Bengali":results['be']['total_num_sents'],"Hindi":results
```



In [12]:
```
for ln in pos_tag_counts:
    for key in pos_tag_counts[ln]:
        pos_tag_counts[ln][key] = pos_tag_counts[ln][key]/(results[ln]['total
```

In [13]:
```
keys_to_be_deleted = []
```

In [14]:
```
for ln in pos_tag_counts:
    for key in pos_tag_counts[ln]:
        if key not in pos_tag_counts['hi'] or key not in pos_tag_counts['be']
            keys_to_be_deleted.append(key)
```

In [15]:
```
keys_to_be_deleted = list(set(keys_to_be_deleted))
keys_to_be_deleted.append('$')
keys_to_be_deleted.append('G_X')
print(keys_to_be_deleted)
```

```
['E', 'U', '~', '#', 'null', '@', '$', 'G_X']
```

In [16]:
```
for ln in pos_tag_counts:
    for key in keys_to_be_deleted:
```

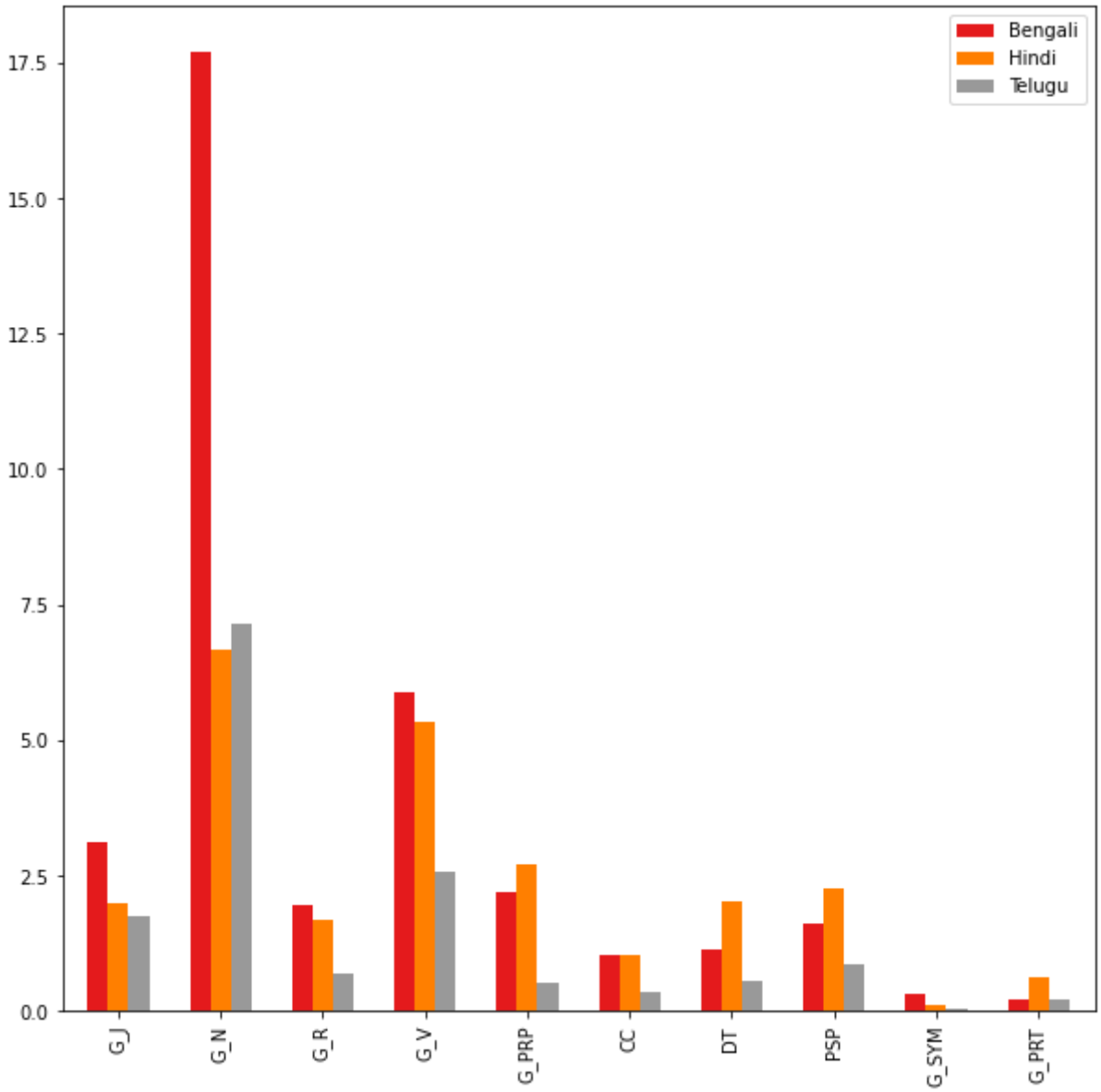
```
        pos_tag_counts[ln].pop(key, None)
```

We have deleted tags that didn't have English words in all the languages, and also Tags that are symbols or Emojis, to get a better understand of the actual data.

We are left with 10 tags, below we take a look at their normalized counts.

In [1]:
```
import pandas as pd
```

In [20]:
```
df = pd.DataFrame({"Bengali":pos_tag_counts['be'],"Hindi":pos_tag_counts['hi'
```



Part of Speech Analysis of English usage in codemixed data

G_J is the Adjective part of speech. We see a higher usage in Bengali, which declines in Hindi and Telugu. It can be reasoned that this is because adjectives are more used in low-codemixed data, where they're used as standalone words. Thus, after accounting for normalization, we see a negative correlation with adjective usage in English and the amount of codemixing. For G_N, which is the noun part, we clearly see that Bengali dominates, for the same reason it does in Adjectives. For small amounts of codemixing, English nouns are readily used. A similar logic also applied to Adverbs (G_R) and Verbs (G_V). Thus, the core parts of speech,i.e, Nouns,

Verbs, Adjectives and Adverbs: all show a negative correlation with amount of codemixing. This is understandable, as they're the ones that can be readily embedded in another language. In terms of most frequent parts of speech, they're the ones being used.

When we get to the second part, of functional parts of speech, we see that the correlation no long holds. In fact, Hindi has higher usage of function parts of speech, which shows higher amount of chunk embeddings. We will look at some examples now:

"accha i should sleep , kaal office hai :(" - In this sentence, we see two chunks, one in English and Hindi. This is a perfectly normal usage, and while "accha" is in Hindi, in this sentence it is a figure of speech, and is being used to start the sentence.

"insti politics koi nayi baat nahi hai bhai , my sympathies wid u" - This is similar, where we see a phrase "My sympathies are with you" used in a codemixed sentence.

"best confession till now ... chinta sudhu aktai , eta sotti confession naki keo troll korlo ????" This is an example from Bengali, where it starts from en English sentence, but the three dots signify the change in language, while there is some amount of codemixing (confession, troll) still present.

" classic joke . e bhabei entertain kore jao ." similar to the above, where a common phrase ("classic joke") is used in a larger sentence

" teaser rakamunde how antey em chepatam wait and see pawan mania", in this Telugu sentence we can see how the codemixing tends to get embedded, and otherwise used as Noun and Verbs. "Wait and see pawan mania" is arguably a different sentence, even if in the same tweet.

"Hero : Dialogue delivery adiripoyindi , action kuda bane chesadu .." This is lesser amount of codeswitching, and more codemixing where some words are from English.

Looking at these forms of analysis, we can possibly use some of this data to come up with rules for generating codemixed sentences. These rules need not be limited to POS tags, as we see some level of uniformity and mostly embeddings in the original sentence. We will explore this in the second part of the project, taking Hindi as the language because that was the only language all of us had native proficiency in.