INSTABASE

Text Classification

Akshita Jajoo
IIT Kharagpur

<Encipher/>

Coding the next generation of automation applications

Data Preprocessing and Feature Engineering

Step 1: Data Cleaning

- Lowercasing and removing unnecessary characters (symbols, whitesapces etc.)
- Stop words removal
- Label Cleaning

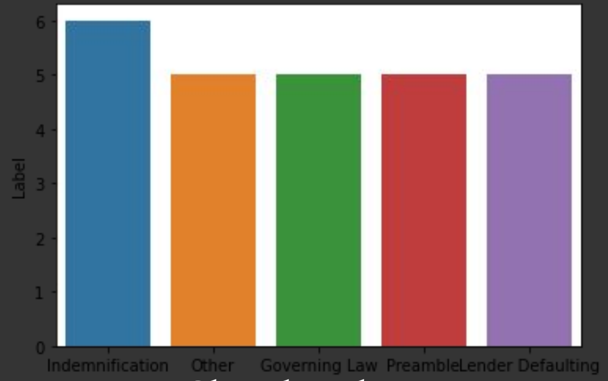
Step 2: Augmentation

- ContextualWordEmbsAug
- Synonyms
- Back Translation

Step 3: Tokenization and Lemmatization

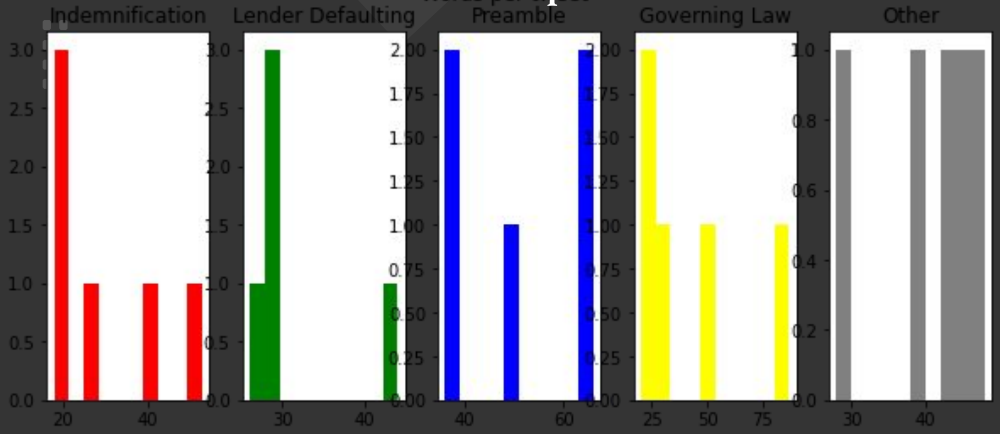
- CountVectorizer
- WordNetLemmatizer

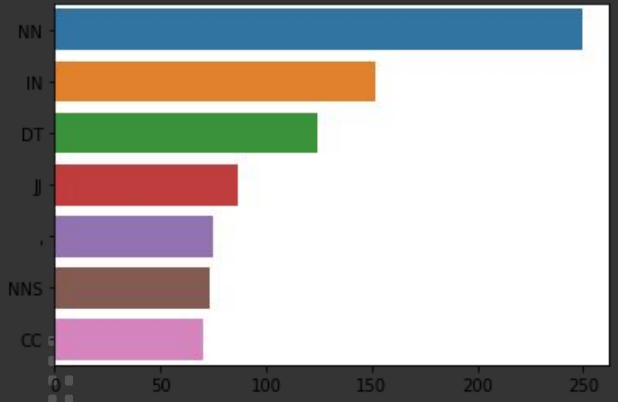
Data Visualization



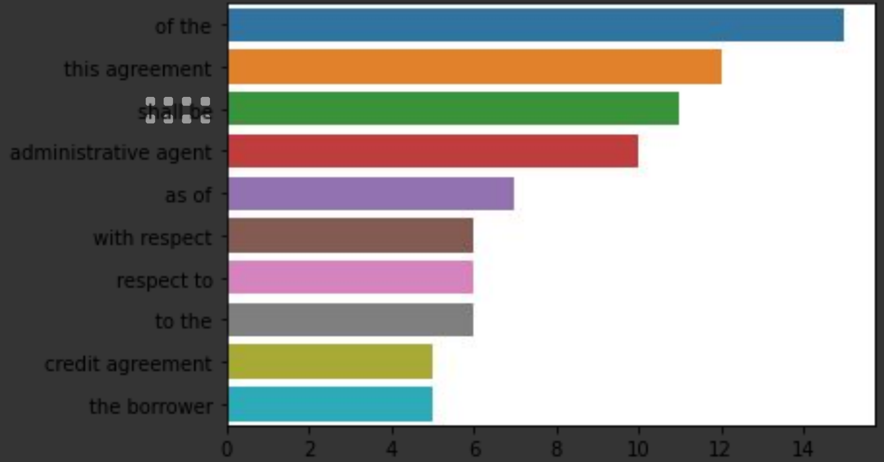
Class distribution

No. of words per label





POS



Top bivariate n gram

Model Options Considered

- ***Baseline Model:*** Naive Bayes (Probabilistic model)
    - Unable to handle new words in test set, gives standard probabilities
- ***Incremental model choice:*** Linear SVC
    - Decent performance with sparse bag of words model
- ***Final Model choice:*** Random Forest Classifier
    - RF classifier was chosen for generalized learning and in case of scaling, word2vec embeddings would perform better while handling unseen words

Reason for not incorporating Neural models

Lack of required data to either train or perform transfer learning on a pre trained model

Final Model Approach

- **Legal clause data**
- Exploratory data analysis and pre processing
  - Lowercasing text,
    Removal of whitespace, punctuation etc.
    POS tagging, Top n grams
- Data Augmentation (as data set is small)
  - Back translation, Contextual word embedding, synonyms
- Tokenization and Vectorization
- Model selection

Model Metrics

Model is Overfitting

Due to insufficient data, our model seems to overfit
With abundant data, our pipeline is set to generalize well

```
Confusion Matrix:
 [[1 0 0 0 0]
 [0 4 0 0 0]
 [0 0 2 0 0]
 [0 0 0 2 0]
 [0 0 0 0 2]]
Accuracy: 1.0
Kappa score 1.0
```

	precision	recall	f1-score
Governing Law	1.00	1.00	1.00
Indemnification	1.00	1.00	1.00
Other	1.00	1.00	1.00
Preamble	1.00	1.00	1.00
accuracy			1.0

Further Improvements

- We can use Word2vec as produces one vector per word is great for digging into documents and identifying content and subsets of content.

- Since our dataset is small we can then use use Logistic Regression after word2vec and the tfidf technique as it has large unsupervised corpus and for each word in the corpus, we try to predict it by its given context

- Pre trained Models like BERT can also be used if we have a larger dataset