Report for Assignment 2

Assignment Task 1

Identifying Generative or Discriminative Models with justification

Neural Networks:
They are Discriminative.
Neural Networks take single input and produce single output
They work on Optimizing only 1 single objective function
They don't learn the underlying distribution of the data

Naive Bayes Classifier:
It is Generative.
It models the full joint distribution P(X,Y)

Logistic Regression:
It is Discriminative.
It models only the conditional distribution,
P(Y | X) and not the full joint distribution P(X,Y)

Gaussian Mixture Model:
It is Generative.
We know that an HMM is a generative Model.
Each and every unit of an HMM is a GMM,
i.e., GMM can be thought of as a single unit HMM,
so, they try to learn the distribution.

GANs:
They are Generative.
They try to learn the true distribution
So that they can generate New and Novel Data Points by sampling

LDA(Latent Dirichlet Allocation):
It is Generative.
It is used for more than 1 task such as
Classification and Dimensionality Reduction
This is possible only if it learns the underlying distribution

SVM:
It is Discriminative.

They try to learn the boundary for each class instead of the distribution
They try to optimize a single objective function such as loss function for classification

Decision Tree:
It is Discriminative.
They try to learn the boundary for each class instead of the distribution
They try to optimize a single objective function such as loss function for classification

Note:
In General,
Generative Models have a single Input and Multiple Outputs
which can be used for other purposes such as classification
and Dimensionality Reduction, etc.
While,
Discriminative Models have a single input and single output
and work on specific problems like classification.

Assignment Task 2

Built the Probabilities from the Frequencies with Additive / Laplacian Smoothing implemented for Emission and Transition Probabilities with k = 0.001

Test Sentences are the final 10 Sentences

Accuracy of the HMM Model with the Viterbi Algorithm :
0.9386653882119602
Confusion Matrix :

	precision	**recall**	**f1-score**	**support**
.	0.943	1.000	0.971	33
X	0.250	0.667	0.364	3
ADJ	0.938	0.833	0.882	18
ADP	1.000	0.963	0.981	27
ADV	0.800	0.889	0.842	9
VERB	0.941	0.914	0.928	35
DET	1.000	1.000	1.000	33
CONJ	0.875	1.000	0.933	7
NOUN	0.978	0.863	0.917	51
PRON	1.000	1.000	1.000	12
PRT	0.917	1.000	0.957	11
NUM	0.000	0.000	0.000	0
avg / total	**0.950**	**0.933**	**0.939**	**239**

Assignment Task 3

The Cooked-Up Features:

- 'bias': 1.0,
- 'word': word,
- 'is_first': i == 0,
- 'is_last': i == len(sent) - 1,
- 'is_capitalized': sent[i][0].upper() == sent[i][0],
- 'is_all_caps': word.upper() == word,
- 'prefix-1': word[0],
- 'prefix-2': word[:2],
- 'prefix-3': word[:3],
- 'suffix-1': word[-1],
- 'suffix-2': word[-2:],
- 'suffix-3': word[-3:],
- 'prev_word': pword(i,sent),
- 'next_word': nword(i,len(sent),sent),
- 'prev_tag': ptag(i,sent),
- 'next_tag': ntag(i,len(sent),sent),
- 'has_hyphen': '-' in word,
- 'is_numeric': word.isdigit(),
- 'capitals_inside': word[1:].lower() != word[1:]

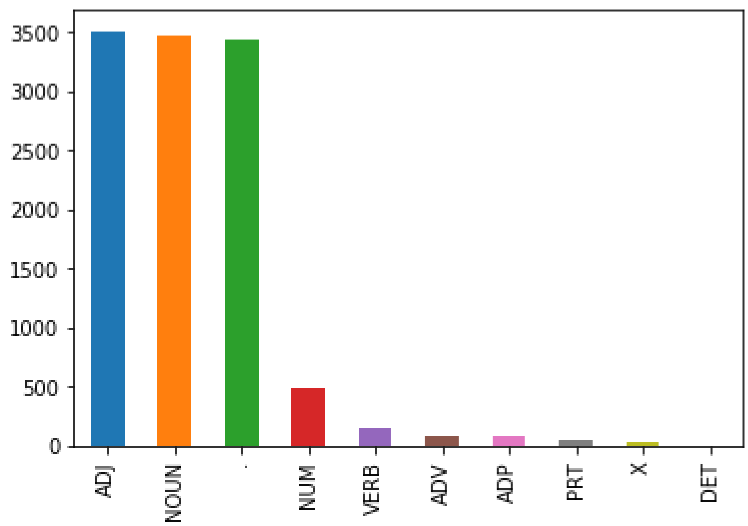
Accuracy of the CRF Model :
0.9831842310598059
Confusion Matrix:

	precision	**recall**	**f1-score**	**support**
.	1.000	1.000	1.000	33
X	1.000	1.000	1.000	3
ADJ	1.000	0.944	0.971	18
ADP	0.963	0.963	0.963	27
ADV	0.900	1.000	0.947	9
VERB	1.000	1.000	1.000	35
DET	0.971	1.000	0.985	33
CONJ	1.000	0.857	0.923	7
NOUN	1.000	1.000	1.000	51
PRON	1.000	0.917	0.957	12
PRT	0.917	1.000	0.957	11
NUM	0.000	0.000	0.000	0
avg / total	**0.984**	**0.983**	**0.983**	**239**

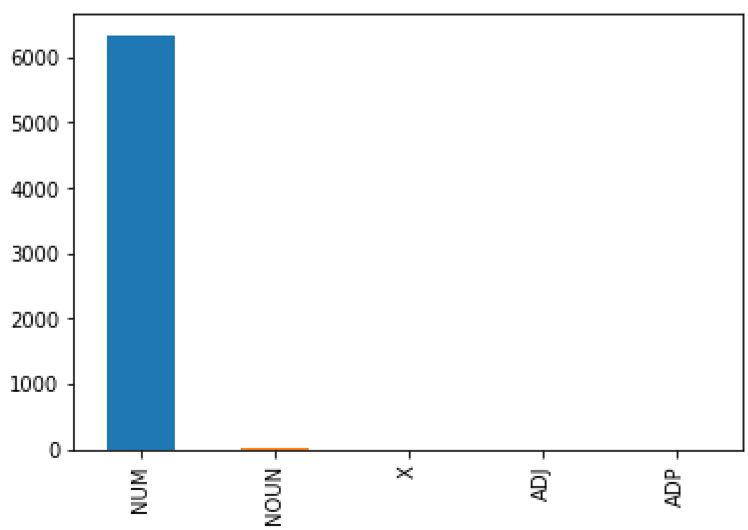
Justification for considering features of CRF

While plotting Histogram of frequencies of features with respect to Tags, if we get Skewed Distributions, It implies that the particular feature is a very good discriminator with respect to tags. The more skewed the distribution is, the better it is.

 # Justification for has-hyphen parameter

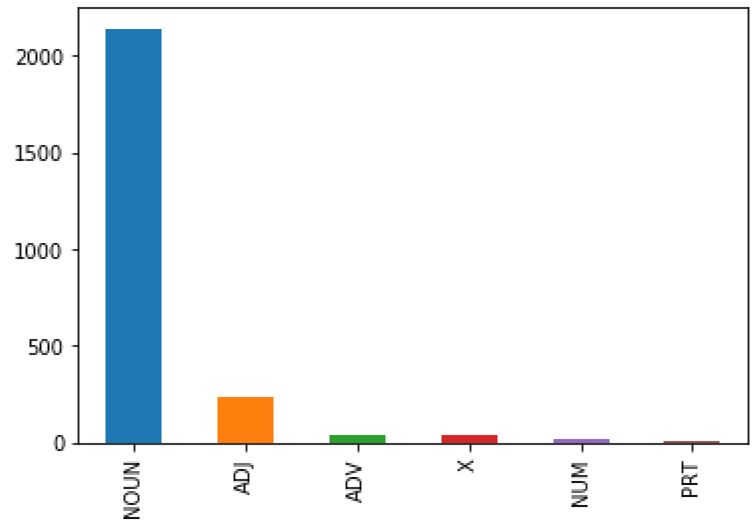


 # Justification for is_numeric parameter

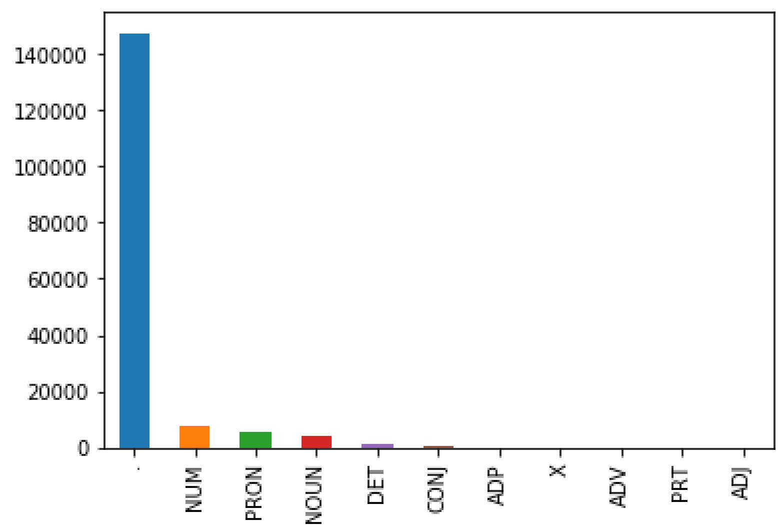


 # Justification for the inclusion of Prefix and Suffix Features
the 2-letter suffix is a great indicator of past-tense verbs, ending in “-ed”.
the 3-letter suffix helps recognize the present participle ending in “-ing”.
Such Similar Patterns exist in prefix too

Justification for capitals_inside parameter



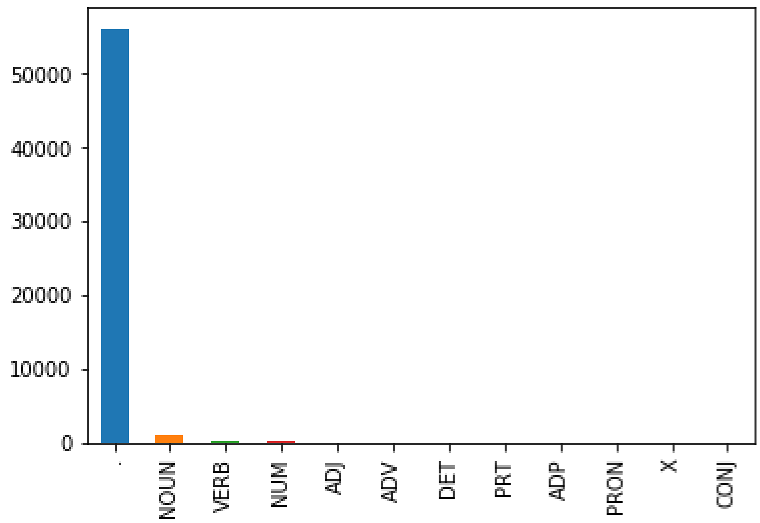
Justification for is_capitalized / is_all_caps parameter



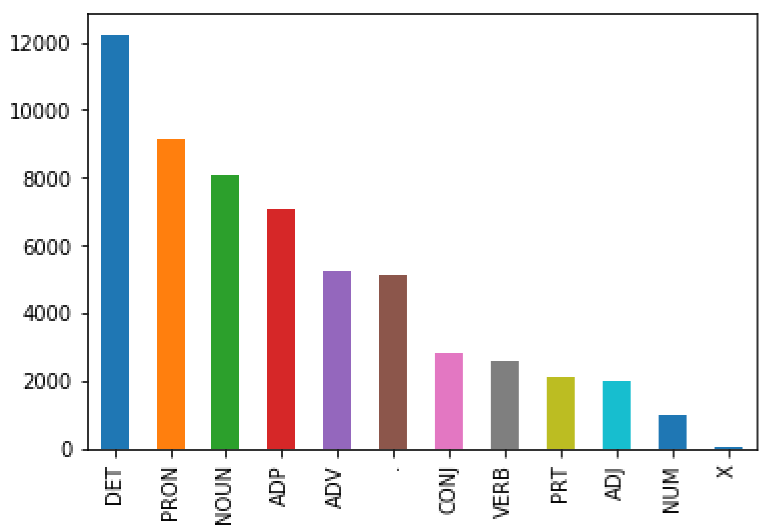
 # Justification for the inclusion of Previous and Next Word :
Most obvious choices for features are: the word itself, the word before and the word after.

 # Justification for the inclusion of Previous and Next Tag:
This choice of features was inspired from the HMM Model where Hidden States are the Tags
And the transitions occur between the Tags (i.e. the Hidden States)

Justification for is_last parameter



Justification for is_first parameter



This distribution isn't as skewed as the ones we saw above, but still the inclusion of this feature leads to increase in accuracy slightly, this might be because this feature in tandem with other features is a better discriminator rather than the features separately.