Urdu Author-disambiguation

Choice of Data

- For this problem I choose urdu afsana nigar disambiguation
- 5 authors was chosen:
    - Devendra Satyarthi
    - Manto
    - Premchand
    - Qurratulain hyder
    - Rajinder singh bedi
- All data are taken from https://www.rekhta.org/
- I wrote a web crawler for data collection

Web crawler

- I wrote a simple web crawler in node.js for gathering afsanas
- This web crawler opens each afsana page after 20 seconds time interval
- Scrape data into a txt file
- Output folder contains folders for each 5 authors
- And each folder contains 20 afsanas txt files of each author
- In each file a complete afsana is on a single line
- These afsanas will become rows of our feature vector
- https://github.com/FahadQurashi49/scraper

Data cleaning

- I removed all punctuation marks, using node.js from the afsanas
- A simple regex was made:
  - **[.\،\ٔ\!\؟\-\’\:\‘\]**
- I also removed stop words all the afsanas, A class CleanAfsanas in python does this
- All stop words are taken from: https://github.com/stopwords-iso/stopwords-ur

Tfidf vectorizer results

- First I used only one feature vector of Tfidf with tri-gram model
- And max features 6000
- Features vector shape is 98 x 6000, (98 afsanas)
- Split data in 75% train and 25% test data sets
- And calculate accuracy of three classifier:

Classifier	Accuracy
MultinomialNB	0.64
SVM_SVC	0.76
XGBClassifier	0.72

Add POS Tfidf as feature vector

- I used another feature vector Tfidf of each author’s part of speech tags
- First of all a dictionary of POS and urdu words are made using the provided data set in assignment resources
- Using this POS dictionary all each word in an afsana is being replaced by its POS tag
- Means now the dataset contains strings of POS tags instead of words
    - 'AP NN NN NN VB VB NN P VB PN NN NEG AA……’
- This data set is the input of tfidf vectorizer

Classifier Pipeline

- A pipeline is made for the algorithm
- In this first afsanas data is cleaned with stop word
- Then this cleaned data is fed to union of features
- Union of features contains union of POS tagger Tfidf and word Tfidf feature vectors
- Then the data is split into 75% train, and 25% testing sets
- Then the classifiers are run and accuracy recorded

Union of features results

- Using Tfidf feature vectors with min_df (minimum document frequency) and max_df(maximum document frequency) also gave some extra boost my model

Classifier	Accuracy
MultinomialNB	0.84
SVM_SVC	0.96
XGBClassifier	0.76