**Data Parsing and Pre-process**

Done..scraping.py, tokenStemRemove.py

**Feature Extraction**

1. **Bag of words ( uni-gram)**

So far TfidfVectorizer is used successfully to extract features from all\_plays\_tokenized.txt to form bag-of-words, the extracted features are arrays of similarities among plays, acts within each play and scenes within each play, those arrays are built based on the frequency use of words. The features of each array is limited to 5000 max.

TfidfVectorizer uses a in-memory vocabulary (a python dict) to map the most frequent words to features indices and hence compute a word occurrence frequency (sparse) matrix. The word frequencies are then reweighted using the Inverse Document Frequency (IDF) vector collected feature-wise over the corpus.

Latent Semantic Analysis is applied then to reduce the dimensionality from 5000 to 2, however the explained variance of the LSA reduced model is about 5% to 10% only.

The extracted features are exported as txt files in folders, so clustering can be done by importing those extracted features arrays, instead waiting for the extraction completion on the raw all\_data.txt fiel, which process can take minutes.

1. **N- gram**

2-grams, 3-grams, 4-grams of words features are extracted in vectors

Completed, see folder N-gramFeatures

1. **Type Token Ratio( Richness of vocabulary)**

<http://www.lexically.net/downloads/version5/HTML/index.html?type_token_ratio_proc.htm>

The type-token ratio V/N, V representing the size of the vocabulary of the sample, and N the number of tokens, is a measure indicating the vocabulary richness of an author.

Completed, see folder TypeTokenRatioFeatures

1. **Readability**

<https://en.wikipedia.org/wiki/Coleman%E2%80%93Liau_index>

The Coleman–Liau index is calculated with the following formula:

CLI = 0.0588{L} - 0.296{S} - 15.8\,\!

The Coleman–Liau index is a readability test designed by Meri Coleman and T. L. Liau to gauge the understandability of a text. The Coleman–Liau index was designed to be easily calculated mechanically from samples of hard-copy text. Unlike syllable-based readability indices, it does not require that the character content of words be analyzed, only their length in characters. Therefore, it could be used in conjunction with theoretically simple mechanical scanners that would only need to recognize character, word, and sentence boundaries, removing the need for full [optical character recognition](https://en.wikipedia.org/wiki/Optical_character_recognition) or manual keypunching.

Completed, see folder Style\_Features, column 4 is the CLI score

Besides CLI score, number of sentences, words, letters, avg lengths of words, avg lengths of setences, are also calculated and exported, there sure will be some multicollinearity among them, a PCA application would be a good pre-processing method

1. **Distribution of word length**( optional, if time allows)

The distribution of words of different lengths has been used as a feature in authorship attribution studies (Diederich et al., 2000). Words with a length of 15-19, 20-24 and 25+ were combined in separate categories.

1. **Distribution of frequent function words**( optional, if time allows)

Traditional approaches to stylometry research use content words rather than function words, assuming that the latter occur to frequently to be of any relevance for style. Nevertheless, function words (e.g. determiners, conjunctions, prepositions) are not under the conscious control of the author and therefore meaningful forstylogenetic studies (Holmes, 1994, 90-91).

1. Emotions frequnecy of ? !

**Clustering**

K means clustered Bag of words features extract( vectorized and LSA,PCA dimensionality reduced) generate the following Outliers plays( when set cluster numebr to 2), and the scenes, acts outliers are also mostly coming from the following plays, which validates consistency

['the first part of king henry the fourth']

['the second part of king henry the fourth']

['the life of king henry the fifth']

['the first part of king henry the sixth']

['the second part of king henry the sixth']

['the third part of king henry the sixth']

['the life of king henry the eighth']

['the life and death of king john']

['the life and death of richard the second']

['the life and death of richard the third']

Useful links :

<http://scikit-learn.org/dev/modules/feature_extraction.html#text-feature-extraction>

<http://scikit-learn.org/dev/auto_examples/text/document_clustering.html#example-text-document-clustering-py>

<http://scikit-learn.org/dev/auto_examples/applications/plot_out_of_core_classification.html#example-applications-plot-out-of-core-classification-py>

<https://www.kaggle.com/c/word2vec-nlp-tutorial/details/part-1-for-beginners-bag-of-words>

<http://scikit-learn.org/stable/modules/clustering.html> ( all clustering methods in sckit)

<http://www.r-bloggers.com/text-mining-the-complete-works-of-william-shakespeare/> R dealing with shakespear and visualization

To-do:

1. ~~Test to the effect of apply PCA to extract features and reduce dimensionality , and compare with the k clustering results with LSA reduced dataset~~
2. ~~Use GMM which works better than K means in dealing with long stripped clusters~~

~~3. Verify the consistency of the clustering results: outlier plays contain outlier acts containing outlier scenes~~

4. Add label text files to denote the row in feature files coming from which play, so the Homogeneity, Completeness, V-measure can be calculated between labels of original and km.labels ?

5. Somehow there are dtype error using HashingVectorizer and TfidfTransformer these two feature extraction methods—open to fix and try if these 2 methods will work better than TfidfVectorizer

6. Self Organized Map? Hierachical Clustering& ~~More features other than bag of words?~~

http://www.clips.ua.ac.be/~kim/Papers/LDV06.pdf