Homework – 3

Name: Prasad S Pande

Assignment Title: Sentiment Analysis Using Maximum Entropy Classifier

Introduction:

- Experiment focuses on the implementation of the **Sentiment Analysis System** for **movie reviews** data presented in **nltk** library.
- Classifier used for this experiment is **Maximum Entropy Classifier.**

Experiment Setup:

- Install Python 2.7 (32 bits).
- **NOTE:** 64 bits Python don't have numpy 1.6.1 compatibility.
- Install nltk library for Python 2.7 (32 bits).
- Install numpy 1.6.1
- Before running the code, make sure that Visual C++ compiler is installed in order to avoid run-time errors.
- IDE used – PyCharm Community Edition 4.0.4

Experimented on following Cases:

- Filtering the stop words from the movie review text.
- Filtering Punctuation characters.
- Using Lemmatization on Corpus data.
- Use of unbalanced positive and negative features from the list.

Result Table:

- **NOTE:** Number of Iterations for each case are taken as an input from the user.
- For the following results, number of iterations performed=**3**

	Without Any text filtering	**Filtering Stop words**	**Filtering Punctuation Characters**	**Using Lemmatization**	**Unbalanced positive and negative feature lists**
Amount of positive and negative features in Training set	**pos-** 3/4th of total positive set **neg-**3/4th total negative set	**pos-** 3/4th of total positive set **neg-**3/4th total negative set	**pos-** 3/4th of total positive set **neg-**3/4th total negative set	**pos-** 3/4th of total positive set **neg-**3/4th total negative set	**pos-** 3/4th of total positive set **neg-**1/4th total negative set
Accuracy	72.2%	71.6%	71.8%	72.2%	25%

Code Description:

- ***classifier=maxent.train(train_toks=training_data,algorithm='GIS',max_iter=int(max_iter))***
  - Train function is used to train the given training sample data using Maximum Entropy Model
  - Train function has the following parameters
    - ***train_toks*** –sample training data based on which maximum entropy classifier is trained.
    - ***algorithm***-type of algorithm used to train the classifier. Default is IIS (Improved Iterative Scaling). We used GIS (Generalized Iterative Scaling) in our case.
    - ***max_iter***-maximum number of iterations used for the given experiment of training data. Default number of iterations are 10.

- ***nltk.classify.util.accuracy*(classifier,test_data)**
  - Accuracy method is used to evaluate the model for the given trained classifier over test data.

Observations and Conclusion:

- Accuracy of the model is directly proportional to the number of iterations used to train the model. As the number of iterations increases the accuracy of the model increases.
- Presence of unbalanced data for positive and negative polarity classes in the training set hamper the accuracy of the model greatly. As shown in the table use of unbalanced polarity data reduces the model accuracy to 25% as compared to the 72.2% in case of normal case without any filtering.

- Stop words are highly common and low information words. As we increase the list of relevant stop words in the corpus, accuracy of the model should get increases ideally. Elimination of the irrelevant stop words might affect the accuracy of the model.
- To extract more relevant features from the training set or to enhance feature extraction function, we can overload the default feature extractor functions used in the Classifier function. A feature extractor is simply a function that takes an argument as an input text and returns a dictionary of features.