Python Natural Language Processing

{ Text normalization:

* Tokenizing (segmenting) words 🡪 Splitting into words
* Normalizing word formats 🡪 Case folding (all to lowercase? Case may be useful for sentiment, stemming)
* Segmenting sentences

Dataset: sentiment analysis dataset (dataset.zip, on blackboard). It includes a training

set (train.csv), a development set (dev.csv), and a test set (test.csv). Each tweet has a

sentiment label (Positive, Negative, Neutral). In the test.csv file, the correct sentiment

label of each tweet has been replaced with a symbol “?”.

* Apply NLP analysis methods of each linguistic level:
* Morphology (Lemmatization, review different types of stemmer see which is better)
  + Formed of morphemes.
  + Two types of morphemes, lexical and grammatical.
  + Stemming is the simplest form of morphological processing.
  + Stems are the base of an English word, which can be surrounded by secondary morphemes called affixes.
  + Stemming involves reducing a word to its lemma.
  + http://www.nltk.org/howto/tokenize.html
* Lexicon (Lexical analysis is dividing the whole chunk of text into paragraphs, sentences and words)
* Syntax ( Grammatical structure of sentences, helps to identify sentence structure and relationship between entities)
* Semantics ( <http://www.nltk.org/howto/semantics.html> )
* To process the input text and extract features (positive, negative, neutral)?
* Then use Logistic Regression as the classifier ( <https://www.nltk.org/api/nltk.classify.html> )
* Discuss results and effectiveness of each method for **sentiment prediction** based on the development set
* Give references and citations to the model you used
* Use td-idf weighted uni-gram bag-of-words model as baseline model