Unit 5 Assignment

**ANLY:520-51 (Fall 2016)**

Dean D’souza

H.U. ID: 168424

# **Solutions:**

1. **Read up on one of the language technologies mentioned in this section, such as word sense disambiguation, semantic role labeling, question answering, machine translation, named entity detection. Find out what type and quantity of annotated data is required for developing such systems. Why do you think a large amount of data is required?**

The language technology I chose is Machine Translation. Machine translation is a sub-field of computational linguistics which focuses on the use of software to translate speech from one language to another. [1]

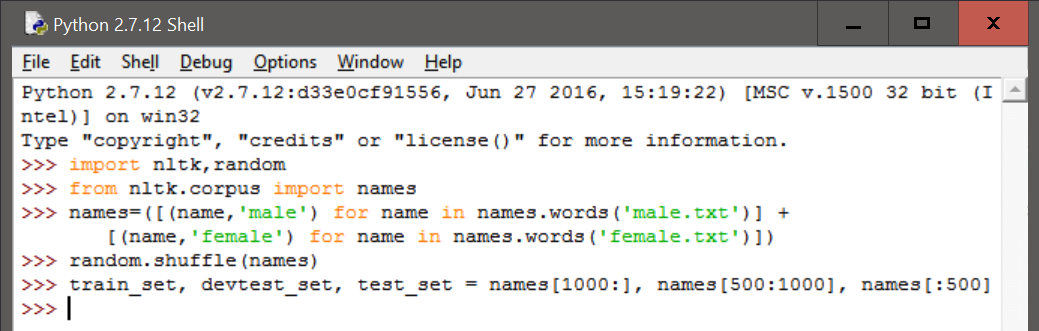
There are a number of ways in which this topic has been addressed but the basic resource required is usually of the form of an extensive lexicon of morphological, syntactic and semantic information and large sets of rules. [1] Some of these methods include:

1. **Rule-Based Machine Translation (RBMT)** which is usually used for translating between closely related languages which usually takes the form of Transfer-based Machine Translation (which uses intermediate representations to try to simulate the original sentence), Interlingual Machine Translation (which uses an intermediate language neutral form to further translate to the target language) and Dictionary-based Machine Translation (which uses a dictionary to translate a word into the equivalent stored in the dictionary).
2. **Statistical Machine Translation** which uses statistical methods combined with text corpora so as to produce accurate translations.
3. **Example-Based Machine Translation** which uses previously translated corpus texts to produce translations which have similar sub-sentential components. The idea behind this is to use analogies to try to convey the meaning of a given sentence.
4. **Hybrid Machine Translation (HMT)** which combines the best features of statistical and rule-based technologies to produce translations. Approaches involve Rules post-processed by statistics, where the text is translated using rules first and then corrected with the help of statistical methods, and Statistics guided by rules, where the Rules for translation are used to both pre and post process the data translated by a statistical translator.

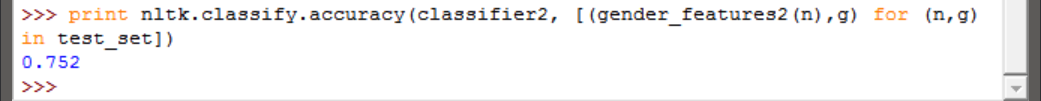
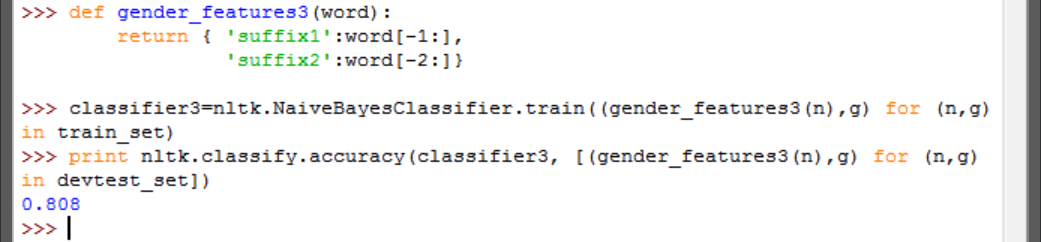
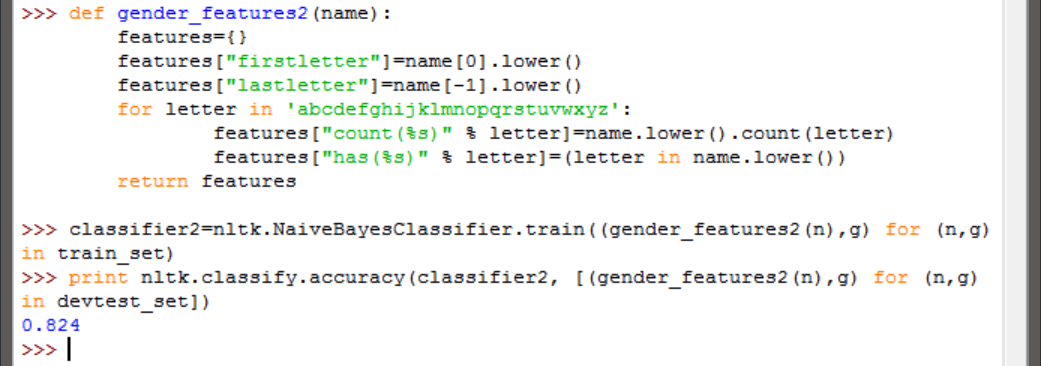
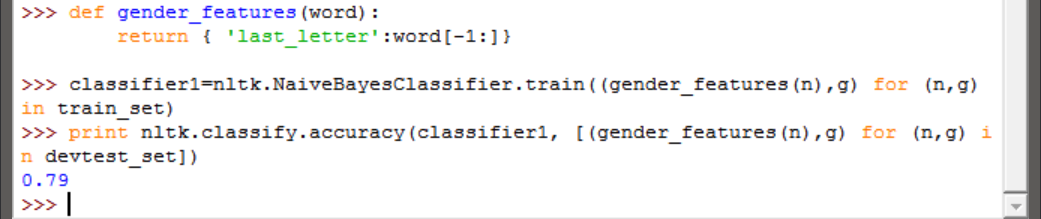
In all the approaches mentioned above, a large amount of data is required due the various intricacies of syntax and semantics contained within each language under consideration, as well as the problems associated with translation rules which requires continuous refinement of such rules for different contexts. A lot of data is required to train the machine to make accurate translations without losing the inherent meaning behind the sentences.

1. **Using any of the three classifiers described in this chapter, and any features you can think of, build the best name gender classifier you can. Begin by splitting the Names Corpus into three subsets: 500 words for the test set, 500 words for the dev-test set, and the remaining 6900 words for the training set. Then, starting with the example name gender classifier, make incremental improvements. Use the dev-test set to check your progress. Once you are satisfied with your classifier, check its final performance on the test set. How does the performance on the test set compare to the performance on the dev-test set? Is this what you'd expect?**

We setup the data as follows:

****

We use the first given definition of gender\_features() to train the classifier and then compare it to the devtest\_set , after which we use the gender\_features() given in the textbook followed by the second version of the gender\_features() and perform the same operations on each.

****

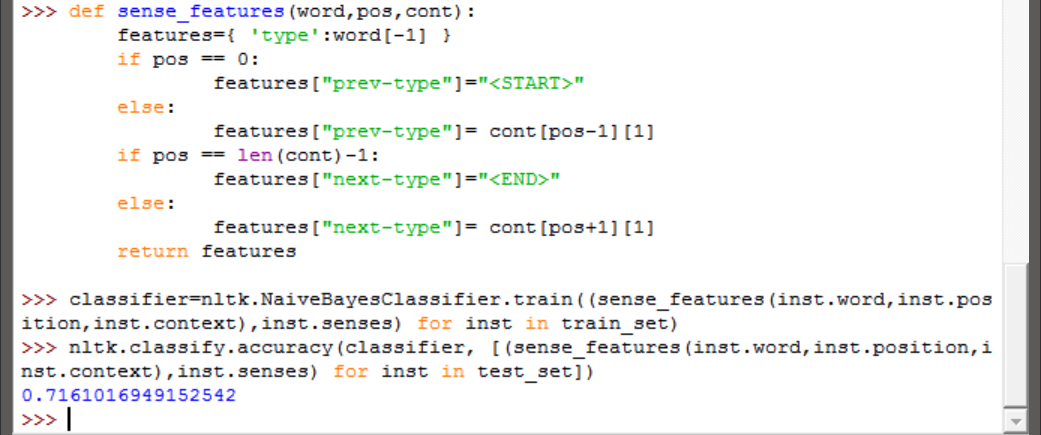
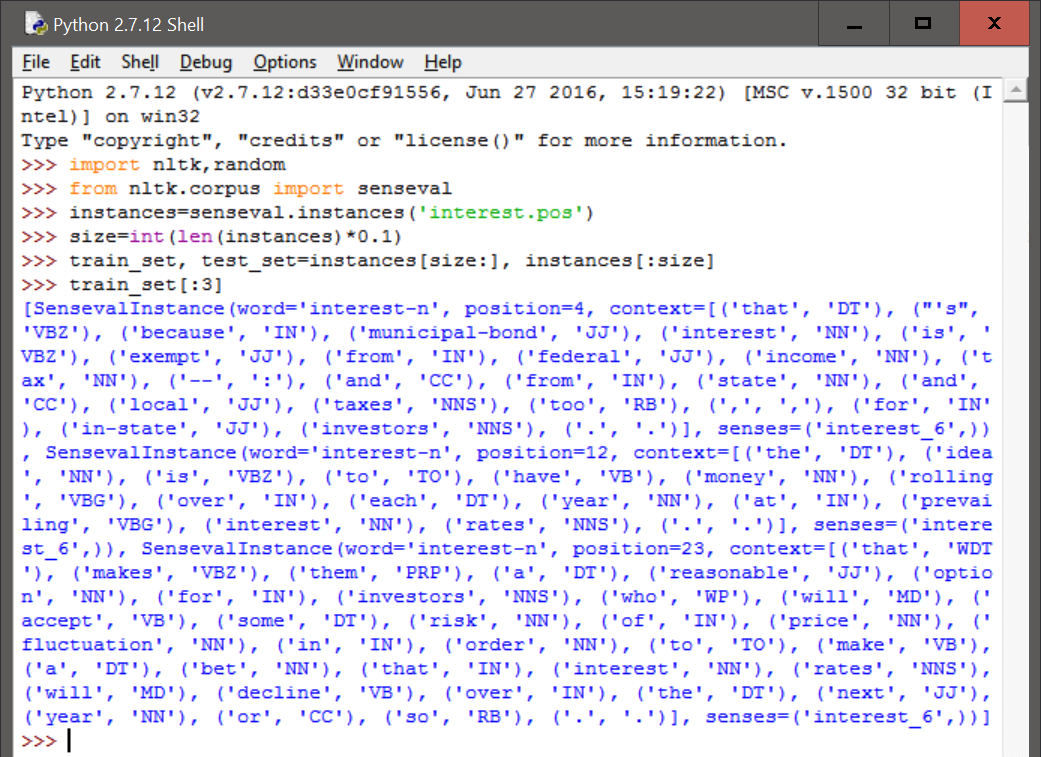
We can see that the gender\_feature2() function gives us a more accurate Naïve Bayes Classifier and use it to test with the test\_set. However, we get a lower value of accuracy which may be due to the test\_data having a better variety of (name, gender) combinations. Hence, it is expected and we can conclude that the classifier constructed needs to be trained better.

1. **The Senseval 2 Corpus contains data intended to train word-sense disambiguation classifiers. It contains data for four words: hard, interest, line, and serve. Choose one of these four words, and load the corresponding data:**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> from nltk.corpus import senseval**  **>>> instances = senseval.instances('hard.pos')**  **>>> size = int(len(instances) \* 0.1)**  **>>> train\_set, test\_set = instances[size:], instances[:size]** | |

**Using this dataset, build a classifier that predicts the correct sense tag for a given instance. See the corpus HOWTO at http://www.nltk.org/howto for information on using the instance objects returned by the Senseval 2 Corpus.**

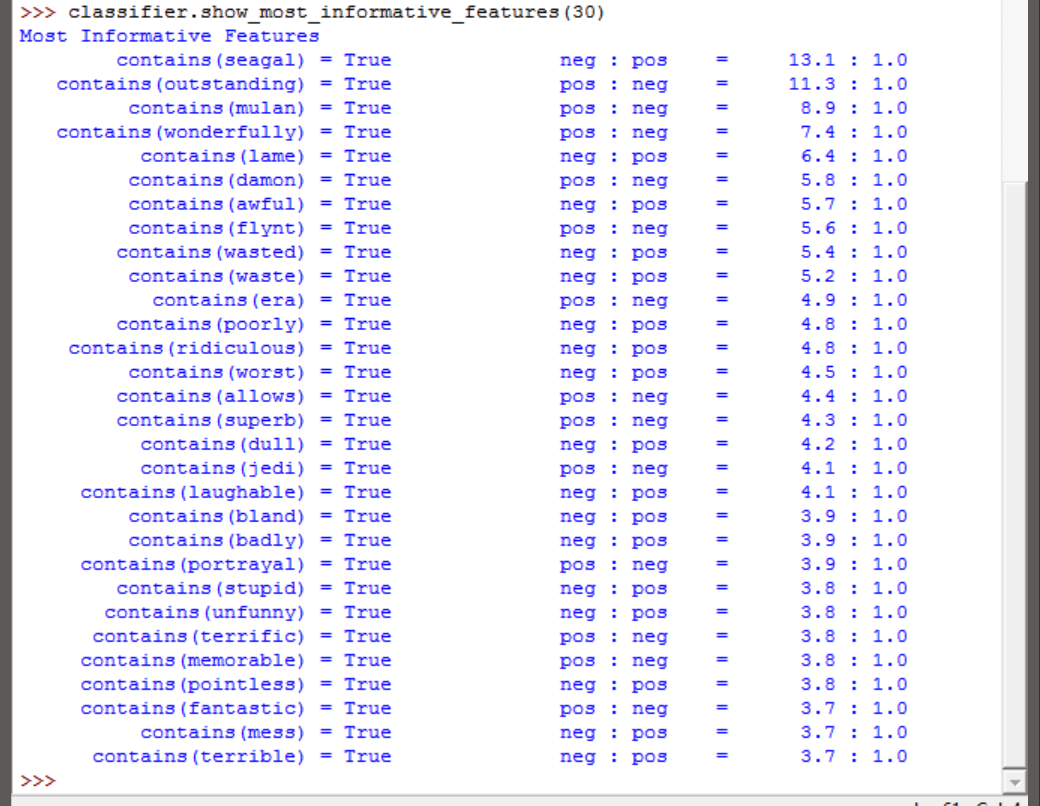
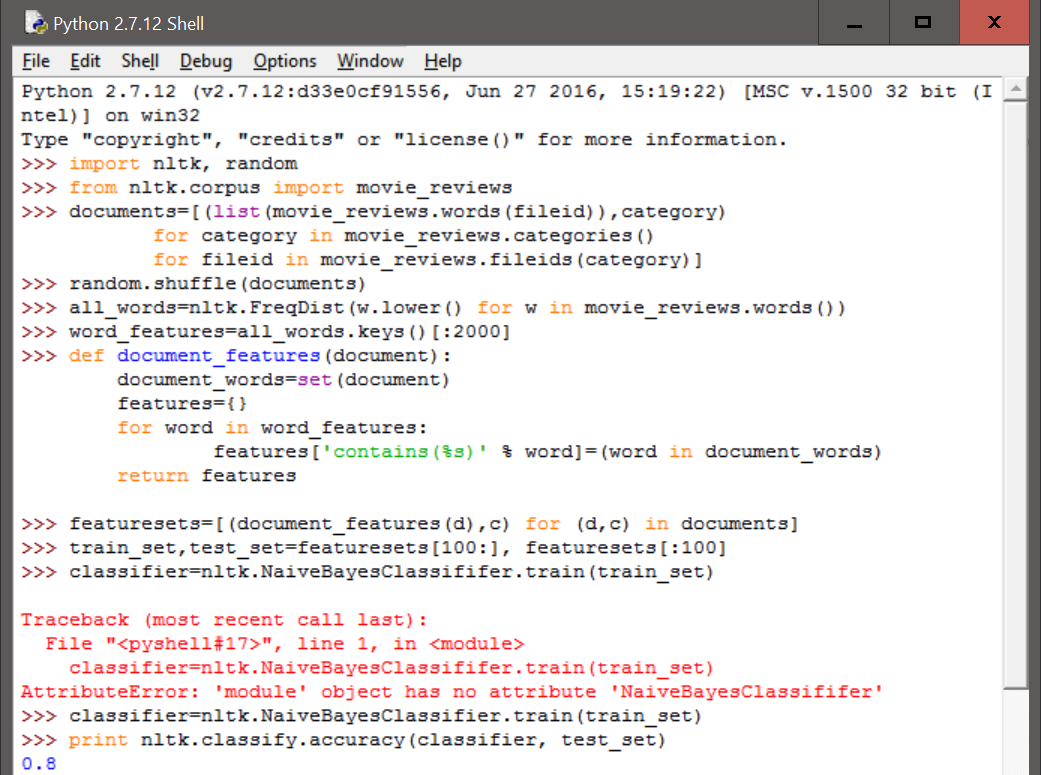
We begin by performing the commands given above and then taking a look at a few objects of the training set as follows:

****

As we can see above, we construct our a sense\_feature function which takes a look at the words last letter, the previous word type and the following word type. This feature function helps us create a Naïve Bayes Classifier which has an accuracy of 71.61% when checked against the test\_set.

1. **Using the movie review document classifier discussed in this chapter, generate a list of the 30 features that the classifier finds to be most informative. Can you explain why these particular features are informative? Do you find any of them surprising?**

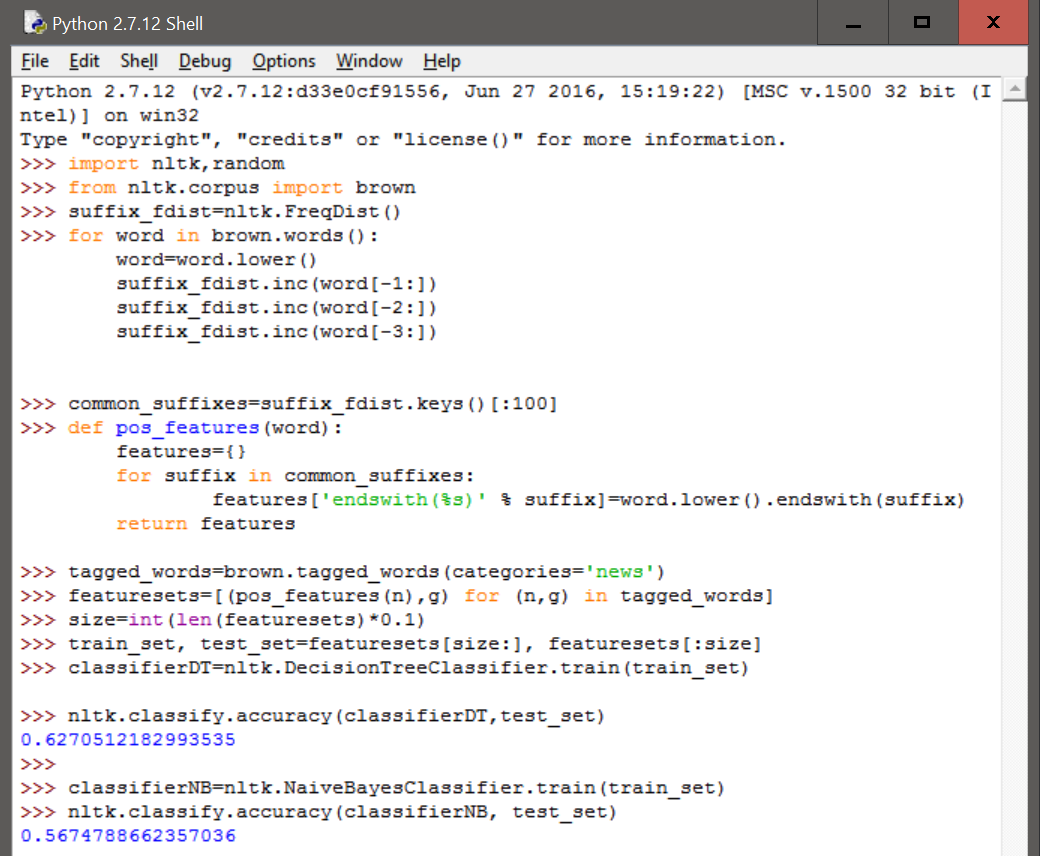
We setup the document classifier from the textbook as follows (we also check the accuracy to account for differences):

****

As we can see in the above list of 30 most informative features, the feature with the most effect on being classified as negative is if the document contains seagal. The other features and their ratios seem to be about right. Though, as a personal opinion, I feel the pos:neg ratio for the feature jedi should be more.

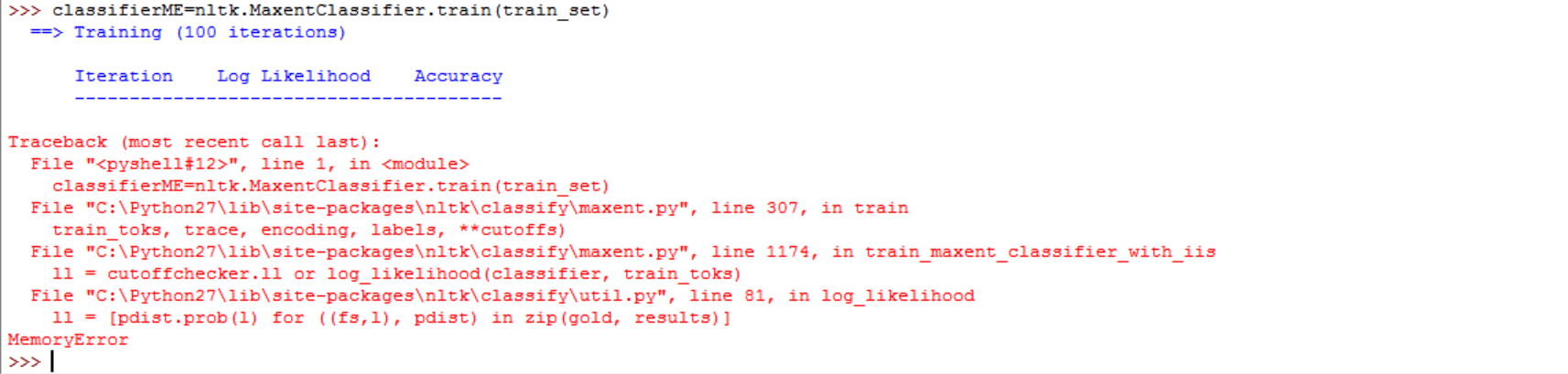
1. **Select one of the classification tasks described in this chapter, such as name gender detection, document classification, part-of-speech tagging, or dialog act classification. Using the same training and test data, and the same feature extractor, build three classifiers for the task: a** **decision tree, a naive Bayes classifier, and a Maximum Entropy classifier. Compare the performance of the three classifiers on your selected task. How do you think that your results might be different if you used a different feature extractor?**

For this exercise I chose the part-of-speech tagging example and performed the required commands as follows:

****

As seen above, when we use the same feature function, a Decision Tree performs better than a Naïve Bayes Classifier in terms of accuracy when compared to the test\_set.

When it came to building a Maximum Entropy Classifier, the program exited with a MemoryError (though my laptop does have 8GB of RAM) as shown below:



The commands were tried twice with the same result. Hence, only a Decision Tree Model and a Naïve Bayes Classifier could be built and compared.

# References:

[1] “Machine Translation”, Wikipedia, (<https://en.wikipedia.org/wiki/Machine_translation>)

[2] “Corpus Readers”, nltk.org, (<http://www.nltk.org/howto/corpus.html>)