INF5830 – 2013, Obligatory assignments, set 2

Part B

To be delivered by Oct. 16, 18:00 (6 p.m.)

Observe, this is the second part of obligatory assignment 2. There is also a part A. Both parts should be completed!

3 Word Sense Disambiguation and Naive Bayes Classification

The staring point is section 6.1 in the nltk book, in particular the subsection called *Document Classification*. You're advised to work through chapter 6 up to and including this section. You should also consider section 6.3 on Evaluation and in particular the subsection on confusion matrices.

a) We will use the famous "line-hard-serve" corpus for word sense disambiguation. It contains training data for one noun, one adjective and one verb. For each, a number of sentences containing the word (from the Brown corpus and elsewhere) has been selected. Each occurrence is tagged with a Wordnet sense.

   Your task is for each of the three words, to construct a Word Sense Disambiguator using the Naive Bayes method. You may use the built-in methods of the NLTK. For each of the three, you should report the accuracy and print a confusion matrix. You should also report a baseline for the classifier.

   You may model much of what you are doing on the subsection on document classification from section 6.1 of the NLTK book. But they are a little sloppy there. They select word features before they split the corpus into training and testing. It is more correct to first split into training and testing (and development) and only use the training corpus for the selection of word features.

   The "line-hard-serve" corpus comes as part of the NLTK data. It is accessed through

   ```
   nltk.corpus.senseval.<method>
   ```

   Try

   ```
   nltk.corpus.senseval.fileids()
   ```

   We may read the relevant part of the corpus for each of the three tasks like

```
>>> l=nltk.corpus.senseval.instances('hard.pos')
>>> type(l)
<class 'nltk.corpus.reader.senseval.SensevalCorpusView'>
>>>
```

But since this is an unfamiliar class, it is more convenient to start the project by reading this into a list which we may manipulate further.

```
>>> l = [i for i in nltk.corpus.senseval.instances('hard.pos')]
>>> random.shuffle(l)
>>> examples_train = l[1000:]
>>> examples_test = l[:500]
>>> examples_dev = l[500:1000]
```

You should also familiarize yourself with the instances in the list (item in the example) and their attributes.

```
>>> item = examples_train[0]
>>> item
>>> item.senses
>>> item.position
>>> item.context
>>> item.word
```

Then you are all set.

b) We will in the following only consider the classifier for "line". Suppose you are only interested in whether it classifies the "product" sense correctly or not. You do not have to construct a new classifier for this, only evaluate how well your current classifier solves this task. What is the recall, precision and accuracy of your classifier considered as a binary classifier in this respect? You may answer this from the confusion matrix. What is the baseline for this classifier?

c) The document classifier in the nltk book has made many choices, we will consider the effect of some of them. We will first consider the effect of the size of the set of word features. They use 2000 words. Try what happens when you use 10, 20, 50, 100, 200, 500, 1000, 2000, 5000 words. We will only make one experiment for each of the set of word features. Here you should use the same training set and test set for all the experiments. You should hand in a table which shows the accuracy results for each choice of size of word features. What do you read from the table?

d) (Optional, but fun:) We have so far chosen the feature words from frequency, but are there smarter ways to choose them? The *classifier.show_most_informative_features()* example in the book shows that some features are more important than others. Now choose your feature words in a smart way and see what you can achieve with 10, 20, 50, 100, 200, 500 words as features.

You may consider what you get with the command *classifier.most_informative_features()*, or you may choose some other association measure.

You should hand in a table with results for the different number of features.

The end