INF5830 – 2015, Obligatory assignments, set 2

To be delivered by Oct. 9, 18:00 (6 p.m.)

1 Word Sense Disambiguation, Naive Bayes Classification and Evaluation

The staring point is chapter 6 in the NLTK book. Before you start working on the assignment, you are advised to work through the NLTK book chapter 6, sections 1.1-1.3 in the 2.ed, (or the corresponding section 6.1 up to *Part-of-Speech Tagging* in the first edition.) In particular, work through the subsection called *Document Classification*. You shold also read section (6.)3 on *Evaluation*. Towards the end ot this exercise, you will also make use of what you learned by solving exercise 3 in exercise set 4 in the weekly exercises (24 Sept.)

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) have been selected. Each occurrence is tagged with a Wordnet sense.

1.1 Part A

Your first 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.

When one does evaluations like this, it is useful to introduce a baseline for the classifier. The baseline is what a straightforward method would have delivered. In general, that is not 0. For accuracy, a baseline method could be $1/n$ where n is the number of classes, or, if you know the majority class, always select it.

To deliver: For each of the three: Accuracy, baseline for accuracy, confusion matrix comparing to the gold test set, the code which you used.

Help to get started: 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 NLTK book uses 2000 features for document classification. To speed up, it sufficed to use 1000 features in exercises $1.1 - 1.5$.

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 you 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
```

1.2 A binary classifier

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. Instead of constructing a classifier with 6 classes, we can construct a binary classifier with only two classes, *product* and *non-product*. Construct such a binary classifier and calculate its accuracy and print a confusion table.

To deliver: Accuracy, baseline for accuracy, confusion matrix comparing to the gold test set, the code which you used.

1.3 Recall and Precision

What is the recall and precision of the binary classifier?

To deliver: Recall and precision and recipe of how they were computed.

1.4 More Recall and Precision

The original classifier using six classes could also be considered a binary classifier for each class. To consider it a binary classifier for the class *product*, we merge the five other classes. Use the confusion matrix from exercise 1.1 to calculate the accuracy, recall and precision for the class *product* of the 6-class classifier in this way. Do the same for the class *cord*

To deliver: Accuracy, recall and precision for the two classes.

1.5 Cross-Validation

For this part, use the 6-class classifier. We have so far split the example set into two sets, one for testing and one for evaluation. You should now try to evaluate by 10-fold cross evaluation instead. Split the example set into 10 equally sized sets. Run 10 experiments where you use the ten sets in turn for evaluation and the rest of the example set for training. Record the accuracy for each experiment. What is the mean and variance of the accuracy across the 10 experiments?

To deliver.: Accuracy for each of the 10 experiments. Mean and variance for the accuracy across the 10 experiments. Code used.

1.6 Feature set size

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 instead use 10, 20, 50, 100, 200, 500, 1000, 2000, 5000 words. Do this first for the classifier which uses 6 classes. We will only make one experiment for each of the set of word features (not cross-validation). Here you should use the same training set and test set for all the experiments to better see the effect ot the feature set independently of the particular test and training set.

Then repeat the same experiment for the binary classifier.

To deliver.: For each of the two classifiers, deliver the accuracy for the different sized feature sets. Code used. Describe shortly what patterns you see.

1.7 Comparing classifiers

Often in language processing we are in a situation where we will evaluate different classifiers against each other. We can illustrate by using the classifiers with different sized feature sets from the last exercise. We will only consider the binary classifier. Using the recorded accuracy as the only data, would you conclude that the classifier using 1000 features is significantly better than the one using 500 features? Would you conclude that the classifier using 1000 features is significantly better than the one using 200 features?

To deliver Answers to the questions with explanations on how you found the answers.

A word of warning To "fish" for significance like this is not considered proper use of statistics. In particular, using .95 as significance level, we are doomed to do mistakes fishing long enough. Still, it is often used in evaluation in language technology.

1.8 Comparing classifiers, part 2

Since we are evaluating the classifiers on the same test set, we have more information available. We can compare two classifiers, $C1$ and $C2$, item for item and record where both are correct, where $C1$ is correct and $C2$ incorrect, where $C2$ is correct and $C1$ incorrect, and where both are incorrect. Do this for the binary classifier using 1000 features and the one using 500 features.

To deliver The results in a 2×2-table. Code.

1.9 Comparing classifiers, part 3

On the basis of this, would you conclude that the classifier using 1000 items is significantly better than the one using 500 items?

To deliver Answers to the questions with explanations on how you found the answers.

2 Conditional Frequency Distributions

This is a continuation of exercise 4 from the first obligatory assignment. There you were asked to compare *he/him* to *she/her*. The person who made the exercise became so excited when he/she spotted an interesting pattern that he/she forgot that *her* is not only the feminine equivalent of *him*, but

also of *his*. Maybe the author also was thinking too much in his/her mother tongue.

Anyway, this was not meant as a trap for you, but when he/she realized the mistake, it was a bit late to change the exercise. Also, this may serve as a good illustration of what often happens. We start out with too simplified assumptions, and have to go back and repeat our experiments with more refinements.

2.1 Include *his*

So what can we do? The simplest thing we may do to get a first idea of the distribution is to also count the occurrences of *his*.

To deliver: The absolute frequency of the five words *she, he, her, him, his* in the Brown corpus and how you found them.

2.2 Using tags

We could then repeat our experiment from assignment 1, comparing *her* not to *him* but to *him+his*. But that does not help us in checking the hypothesis that we have formulated, that

> Not only is the masculine pronoun more frequent than the feminine pronoun, but this discrepancy is more prominent when the pronoun occurs as a subject than as an object.

We can check the hypothesis with the help of a tagged corpus if the corpus tags *her* as a personal pronoun differently from *her* as a possessive pronoun. The tagged Brown corpus with the full tag set does that. Use this to count the occurrences of *she, he, her, him* as personal pronouns and *her, his* as possessive pronouns.

To deliver: A 3x2 table with the results. Code and explanation of how you found the results.

2.3 Is the hypothesis correct?

Use that table and a statistical test to answer the question.

I guess this point requires background in statistics that not everyone has acquired at this stage, but I hope everybody is able to solve this exercise after completing the course.

To deliver: Answer to the question with explanation.

The end