NLP Coursework:Text Classification

October 30, 2013

1 To start: No credit section

Use NLTK as you know from the last lab sheet.

Several corpora in NLTK are in some form divided into text types. From these we can learn classifiers for recognising these types and also use another part of these corpora for testing such classifiers. Examples are:

1. The Brown corpus is categorised by genres such as editorial, news, reviews. Try out the following commands to access these genres that allow you to print the categories that are in the corpus access words by categories and count words in certain categories only.

```
>>> from nltk.corpus import brown
>>> print(brown.categories())
>>> news_words = brown.words(categories="news")
>>> print (news_words)
>>> fdist = ntltk.FreqDist(news_words)
>>> fdist.tabulate(10)
```

You can compare genres via their use of different word types, such as modals. An example to try is given below:

```
from nltk.corpus import FreqDist
modals = ["may", "could", "will"]
genres = ["adventure","news","government","romance"]

Now count how often the different modals occur in the genres of interest
for g in genres:
     words=brown.words(categories=g)
     fdist = FreqDist([w.lower() for w in words
                     if w.lower() in modals])
     print g,fdist
```

Is the result what you would expect?

You can achieve the same with conditional frequency distributions which are more elegant.

```
from nltk.corpus import ConditionalFreqDist
cfdist = ConditionalFreqDist()
for g in genres:
    words=brown.words(categories=g)
    for w in words:
         if w.lower() in modals:
             cfdist[g].inc(w.lower())

cfdist.tabulate()
```

2. The Reuters corpus contains texts classified into 90 topics and is already grouped into a training and test set. Unlike in the Brown corpus, texts can be in more than 1 category.

   ```
   from nltk.corpus import reuters
   >>> reuters.fileids()
   [’test/14826’, ’test/14828’, ’test/14829’, ’test/14832’, ...]
   >>> reuters.categories()
   [’acq’, ’alum’, ’barley’, ’bop’, ’carcass’, ’castor-oil’, ’cocoa’,
   ’coconut’, ’coconut-oil’, ’coffee’, ’copper’, ’copra-cake’, ’corn’,
   ’cotton’, ’cotton-oil’, ’cpi’, ’cpu’, ’crude’, ’dfl’, ’dlr’, ...]
   >>>reuters.categories(’training/9865’)
   [’barley’, ’corn’, ’grain’, ’wheat’]
   >>> reuters.fileids([’barley’, ’corn’])
   [’test/14832’, ’test/14858’, ’test/15033’, ’test/15043’, ’test/15106’,
   ’test/15287’, ’test/15341’, ’test/15618’, ’test/15618’, ’test/15648’, ...]
   ```

   Access to words in a specific category etc follows the same way as for the brown corpus.

3. The sentiment corpus contains a corpus of movie reviews, 1000 positive and 1000 negative ones.

   Go to chapter 6 in the NLTK book and work through the following section: 6.1 until the end of Document Classification (before POS tagging starts). The secion has a start on classifying names which is not really text classification but you learn how to construct features in NLTK so work quickly through it but concentrate on the smaller document classification section which uses a binomial NB.

   You should now have a classifier that classifies the movie review corpus using just bag of words, using the 2000 most frequent words as features. Make sure you understand what the relatively concise code is doing.

   It also uses a very large training set and only a small test set. Please change this to using three quarters of the corpus as training and 1 quarter as testing.

2 Text Classification: Credit section

The coursework counts for 10% of the module. You need to submit a single report in pdf format where each section addresses one of the following questions (in order). As some questions also ask for a program, please submit those with a clear name such as *yourname-question1.ext* together with the report in a single zipped file. The section of the report must then also contain a short README on how to run the program and some documentation of it. [1]

1. Using the movie review document classifier, generate a list of the 30 features that the classifier finds to be most informative. Explain why these particular features are informative and which ones might be surprising. Give the command you used with output in the report but a separate program submission is not necessary as the command is given in the NLTK book! [4 marks]

2. You can run your classifier also on invented reviews that you give interactively as lists such as on [“the”, “plot”, “was”, “ludicrous”]. Invent two short reviews that your classifier classifies wrongly and explain why this is. The two reviews should exhibit *different* linguistic properties that are responsible for the mistakes made. No program to be submitted. [6 marks]

3. Investigate the impact of different feature selection methods on the classifier as discussed in the lecture, including feature selection by frequency cutoffs, association measures, and function word

[1] You are allowed to use non-python and non-NLTK as the corpus is available as text online at `http://www.cs.cornell.edu/people/pabo/movie-review-data/`, for example by using your own feature extraction and then WEKA (Weka’s Naive Bayes) as the machine learning toolkit but if you want to do so, please contact me ASAP to discuss. You will need to write feature extraction to convert into Weka format equivalent to the code above. I strongly recommend using NLTK, though.

exclusion (NLTK has a stopword list implemented.) Submit your new feature selector functions as well as a section in your report commenting on the results. You should investigate and explain the results (for example by looking at the stopword list or linking your results to the NB algorithm). You should be systematic in your exploration of different cutoff points. [10 marks]

4. Word features can be very useful for performing document classification, since the words that appear in a document give a strong indication about what its semantic content is. However, many words occur very infrequently, and some of the most informative words in a document may never have occurred in our training data. One solution is to make use of a lexicon, which describes how different words relate to one another. Using WordNet lexicon, augment the movie review document classifier to use features that generalize the words that appear in a document, making it more likely that they will match words found in the training data. There are several ways in which you could do this in WordNet as it includes synonyms, hypernyms as well as similarity measures. Be creative. Submit your new program(s) and discuss results. The WordNet lexicon was introduced to you by Eric in his IR lectures and can be accessed in NLTK by studying Chapter 2.5 in the NLTK book. [10 marks]

5. This question addresses theoretical work and is independent of the movie review dataset! You are given the following documents:

   - D1: He moved from London, Ontario, to London, England
   - D2: He moved from London, England, to London, Ontario
   - D3: He moved from England to London, Ontario

   Which of the three documents have identical and different bag of words representations for (i) the Bernouilli model and (ii) the multinomial model? If there are differences describe them. [4 marks]

6. Another question on theoretical work. Based on the data in the Table below , (i) estimate a multinomial Naive Bayes classifier, (ii) apply the classifier to the test document, (iii) estimate a Bernoull NB classifier, (iv) apply the classifier to the test document. You need not estimate parameters that you don't need for classifying the test document. [6 marks]

	docID	words in document	in c = China?
training set	1	Taipei Taiwan	yes
	2	Macao Taiwan Shanghai	yes
	3	Japan Sapporo	no
	4	Sapporo Osaka Taiwan	no
test set	5	Taiwan Taiwan Sapporo	?

3 Administrative Details

- Worth 10% of the module.
- Deadline: 14.11.2013. 10 o'clock via VLE.
- Typed report of not more than 10 pages, minimum font size 11 points. Pdf format only. Exception: Answer to the last question is allowed to be submitted by orderly hand-written and scanned answer, if easier.
- All work will be monitored for plagiarism.