INF5830, 2015, Exercise 19 Nov

Exercises on Collocations

To study collocations seriously one has to use large corpora. To simplify, we use a corpus where many phrases have a tendency to reappear, the Inaugural corpus supplied with NLTK. We consider the whole corpus.

We first convert the corpus to the class nltk.Text which gives us access to the method Text.collocation(). See the first pages in section 3.1 in the NLTK book.

We want more control and in particular to test out the effect of various association measures as described in chapter 5 in the FSNLP book. NLTK may also provide tools for this in the Collocation package. Unfortunately, when we move outside the NLTK book, the documentation isn't always as easy to follow, but we will try to take it stepwise. You will find some documentation at http://www.nltk.org/howto/collocations.html

We will use two basic tools, and we start by giving them simpler names, so it becomes easier to refer to them

```
>>> Finder=nltk.collocations.BigramCollocationFinder
>>> AssocMeasures = nltk.collocations.BigramAssocMeasures
```

Finder is the basic tool we will use to find the collocations. It is a class with several useful methods. AssocMeasures will provide the various association measures.

Collocations are calculated from the frequencies of words and the frequencies of bigrams. We may build a bigram finder for the Inaugural corpus as follows.

```
>>> inaugural = nltk.corpus.inaugural
>>> tokens = inaugural.words()
>>> word_fd = nltk.FreqDist(tokens)
>>> bigram_fd = nltk.FreqDist(nltk.bigrams(tokens))
>>> inaug_finder = Finder(word_fd,bigram_fd)
```

We may then test our collocation finder and print out the top 20 results.

```
>>> scored = inaug_finder.score_ngrams(AssocMeasures.raw_freq)
>>> scored[:20]
```

Exercise 1

As we see, many of the results are not good candidates. We will therefore do some filtering. We should remove non-words and possibly also stop words. This may done on the model of

```
>>> inaug_finder.apply_word_filter(lambda w: not(w.isalpha()) )
```

You may check how the result has changed:

```
>>> scored = inaug_finder.score_ngrams(AssocMeasures.raw_freq)
>>> scored[:20]
```

You may apply apply_word_filter several times on different criteria to get the best result.

You may also consider removing low-frequent bigrams with the filter

```
>>>finder.apply_freq_filter(<n>)
```

Where you provide a number for <n>

Exercise 2

You are now ready to compare various association measures. In addition to the raw frequency (raw_freq) you should try

- Dice: dice
- T-score: student_t
- Chi squared: chi_sq
- Pointwise mutual information: pmi
- Log-likelihood ratio: likelihood_ratio

Print out the 20 top-ranked collocations for each of the 6 tests as well as for the Text.collocation method we started with.

Exercise 3

As a help to understand the different measures, for each pair, in addition to the score provided, also print the absolute frequency of the word pair and of each of the two words in the pair.

Exercise 4

Of the proposed collocations, give 5 examples you think should classify as collocations and 5 examples you think are not collocations. Discuss how these 10 word pairs relate to the possible criteria for collocations (sec. 5.5 in Foundations of Statistical Natural Language Processing)