NLP For Economists

Corpus Exploration

Sowmya Vajjala

Munich Graduate School of Economics - LMU Munich

Guest Course, October 2020

Goals

- Understand what corpus analysis is and why it is needed
- Do some basic analyses
- source: Chapter 1-2 in nltk.org/book

What is corpus exploration?

- Looking at frequently used words/ngrams in the corpus
- What words go together? (collocations)
- How can we visualize a corpus quickly?
- Other properties of the corpus such as lexical diversity, linguistic coverage etc.

.. ...

Why?

- Understand what the texts are about (generally speaking)
- What can be some useful features to use in an NLP model for this corpus
- Identify potential noise in the data
- Understand the limitations (may be something you want is not represented in the corpus?)

How? - getting some basics covered

I am taking the example texts that come with nltk in this example

```
from nltk.book import * #loads some sample texts
texts() #lists the texts.
list(text1) #shows a text as a list of words.

#corpus of 3 books - I am first getting list of words for each text,
#concatenating them, and forming a new Text object, which is like a corpus in NLTK.
largercorpus = Text(list(text1) + list(text2) + list(text3))
```

What we can do with this corpus:

```
>>> largercorpus.
largercorpus.collocation_list(  largercorpus.concordance(       largercorpus.dispersion_plot(
  largercorpus.index(             largercorpus.readability(       largercorpus.unicode_repr(
largercorpus.collocations(      largercorpus.concordance_list(  largercorpus.findall(
  largercorpus.name               largercorpus.similar(           largercorpus.vocab(
largercorpus.common_contexts(   largercorpus.count(             largercorpus.generate(
  largercorpus.plot(              largercorpus.tokens
```

How? - Frequency Distributions

```
fdist = FreqDist(largercorpus)
```

What we can do with this fdist object:

```
>>> fdist.
fdist.B(              fdist.clear(          fdist.freq(           fdist.hapaxes(
fdist.max(            fdist.plot(           fdist.pprint(         fdist.subtract(
fdist.update(
fdist.N(              fdist.copy(           fdist.fromkeys(       fdist.items(
fdist.most_common(    fdist.pop(            fdist.r_Nr(           fdist.tabulate(
fdist.values(
fdist.Nr(             fdist.elements(       fdist.get(            fdist.keys(
fdist.pformat(        fdist.popitem(        fdist.setdefault(     fdist.unicode_repr(
>>> fdist.most_common(30)
[(',', 31791), ('the', 19993), ('.', 12152), ('and', 11802), ('of', 11459), ('to', 9
216), ('a', 6954), ('in', 6408), (';', 6096), ('that', 4788), ('I', 4612), ('it', 40
67), ('his', 4051), ("'", 3835), ('was', 3795), ('he', 3204), ('"', 2984), ('for', 2
945), ('her', 2938), ('with', 2919), ('-', 2918), ('as', 2911), ('s', 2702), ('is',
2690), ('be', 2589), ('not', 2539), ('all', 2349), ('at', 2090), ('him', 2078), ('yo
u', 1960)]
```

How? - Frequency Distributions of ngrams

```
from collections import Counter
from nltk import ngrams
ngram_counts = Counter(ngrams(largercorpus, 4)) #4 for bigrams, 3 for trigrams ...
ngram_counts.most_common(10)
```

What we can do with this fdist object:

```
>>> ngram_counts.most_common(10)
[(('Mrs', '.', 'Jennings', ','), 74), (('it', 'came', 'to', 'pass'), 66), (('And', '
he', 'said', ','), 65), (('.', 'And', 'he', 'said'), 64), (('And', 'it', 'came', 'to
'), 60), (('.', 'And', 'it', 'came'), 56), (('in', 'the', 'land', 'of'), 53), (('the
', 'ship', "'", 's'), 52), (('the', 'whale', "'", 's'), 49), ((',', 'and', 'said', '
,'), 47)]
```

How? - Collocations

"A collocation is a sequence of words that occur together unusually often. Thus red wine is a collocation, whereas the wine is not. A characteristic of collocations is that they are resistant to substitution with words that have similar senses; for example, maroon wine sounds definitely odd."

`largercorpus.collocation_list()`

```
>>> largercorpus.collocation_list()
['Colonel Brandon', 'Lady Middleton', 'Sir John', 'Sperm Whale', 'Moby Dick', 'said
unto', 'White Whale', 'Miss Dashwood', 'every thing', 'old man', 'Captain Ahab', 'th
ou hast', 'sperm whale', 'Right Whale', 'thou shalt', 'dare say', 'thousand pounds',
 'pray thee', 'Miss Steeles', 'thy seed']
>>> text1.collocation_list()
['Sperm Whale', 'Moby Dick', 'White Whale', 'old man', 'Captain Ahab', 'sperm whale'
, 'Right Whale', 'Captain Peleg', 'New Bedford', 'Cape Horn', 'cried Ahab', 'years a
go', 'lower jaw', 'never mind', 'Father Mapple', 'cried Stubb', 'chief mate', 'white
 whale', 'ivory leg', 'one hand']
>>> text2.collocation_list()
['Colonel Brandon', 'Sir John', 'Lady Middleton', 'Miss Dashwood', 'every thing', 't
housand pounds', 'dare say', 'Miss Steeles', 'said Elinor', 'Miss Steele', 'every bo
dy', 'John Dashwood', 'great deal', 'Harley Street', 'Berkeley Street', 'Miss Dashwo
ods', 'young man', 'Combe Magna', 'every day', 'next morning']
>>> text3.collocation_list()
['said unto', 'pray thee', 'thou shalt', 'thou hast', 'thy seed', 'years old', 'spak
e unto', 'thou art', 'LORD God', 'every living', 'God hath', 'begat sons', 'seven ye
ars', 'shalt thou', 'little ones', 'living creature', 'creeping thing', 'savoury mea
t', 'thirty years', 'every beast']
```

Concluding Remarks

- How to make these more useful/meaningful?
  - Do some corpus pre-processing (e.g., lowecase, remove stop words, punctuation etc)
  - Use plotting functions in nltk documentation for basic visualizations
- Think about what other analyses you can do, and see if there are off the shelf functions in any NLP libraries for these.
- Remember: These are just a few sample analyses.

Note: Go through the first 2 chapters of the NLTK book. It has good examples of doing such exploratory analysis of a corpus.