

Applied NLP Week 10: Lexical Semantics with WordNet

Julie Weeds

November 23, 2017

1 Getting Started

- Open up your preferred Python development environment e.g., Jupyter notebook, Anaconda Spyder or Pycharm.
- You should already have nltk installed. However, depending on your environment, installation of nltk doesn't necessarily download all of the associated resources. To access wordnet you may need to run the NLTK Downloader. You can do this from a python terminal or from a Jupyter notebook cell

```
import nltk
nltk.download()
```

- Go to the Corpora tab and select wordnet and wordnet_ic to download. There are also lots of other interesting things here you may want to play with another time
- Now you should be able to import wordnet into python and use the library functions.

```
from nltk.corpus import wordnet as wn
from nltk.corpus import wordnet_ic as wn_ic
```

2 Useful WN Functions

See <http://www.nltk.org/howto/wordnet.html> for more information (or search for "nltk wordnet") Find out or work out what all of the following functions do.

```
wn.synsets("book")
wn.synsets("book", wn.NOUN)
synsetA=wn.synsets("book", wn.NOUN)[0]
synsetA.definition()
synsetA.hypernyms()
synsetA.hypernyms()
synsetB=wn.synsets("book", wn.NOUN)[1]
synsetA.path_similarity(synsetB)
brown_ic=wn_ic.ic("ic-brown.dat")
synsetA.res_similarity(synsetB, brown_ic)
synsetA.lin_similarity(synsetB, brown_ic)
```

3 Tasks

1. Write a function to return the path similarity of two words. This is generally defined as the **maximum** similarity of all of the possible pairings of the senses of the two words. Make sure you test it. For (chicken, car) the correct answer is 0.0909 (3sf).
2. Generalise it so that you have an extra (optional) parameter which you use to select the WordNet similarity measure e.g., res_similarity and lin_similarity
3. One way to evaluate measures of semantic similarity is to compare the outputs with gold standard similarity judgements produced by human judges. There are a number of standard datasets used in the literature including the MC30 dataset (Miller and Charles, 1991), which

is provided in `mcdata.csv` . Read this file in. Calculate the WN similarity score for each pair using at least 3 semantic similarity measures. Correlate the scores with the human judgements (I suggest you use numpy or scipy for this) and evaluate which is the best measure.

4 Discussion Points

- Were the differences between the measures significant?
- What correlation coefficient did you use? Does it matter?
- For which pairs of words was there most agreement and for which pairs most disagreement?
- What are the problems with using this kind of evaluation of semantic similarity measures?
- To use the `resnik` and `lin` similarity measures, it is necessary to specify a corpus (e.g., `'ic-brown.dat'`). Why? How might this affect the results? What do you need to do to a corpus before you can plug it into a WordNet similarity measure? How might this affect the results?

References

G.A. Miller and W.G. Charles. 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*, 6(1):1–28.