Computational Semantics and Pragmatics 2013

Raquel Fernández – ILLC, University of Amsterdam

Homework #1 Due: 20/09/2013

Submit one PDF file (LaTeX strongly preferred) and one code file for exercise 4 via email.

Exercise 1. Consider the following sentence and the three possible continuations given:

(1) A bicycle was parked near a huge tree in front of an old building.
Its { handlebar | trunk | roof } was strangely shaped.

Humans have no difficulty resolving the reference of the pronoun *'its'* to either *'bicycle'*, *'tree'*, or *'building'* in each case. However, defining a computational procedure that models this ability is far from trivial.

(a) Discuss briefly how the knowledge required to resolve the pronoun appropriately is accounted for in Pustejovsky's Generative Lexicon theory.

(b) How could WordNet be used to design an automatic procedure that would resolve the pronoun appropriately? Use the NLTK corpus reader for WordNet to investigate this question. In your explanation, show the queries that you have used and the output for those queries.

Exercise 2. In their article "Contextual correlates of semantic similarity" publised in the journal *Language and Cognitive Processes* in 1991, the psychologists George Miller and Walter Charles report the results of an experiment where subjects were asked to judge how similar the meaning of a pair of words was on a 5-point scale from 0 to 4. These are the mean similarity scores obtained for 6 noun pairs (out of a total of 30 pairs),[1] with two pairs representing high-level similarity, two pairs representing medium-level similarity, and two pairs representig low-level similarity:

car-automobile	3.92	food-fruit	3.08	monk-slave	0.55
coast-shore	3.70	journey-car	1.16	moon-string	0.08

Use one of the predefined similarity measures in the NLTK corpus reader for WordNet to score the similarity of each of these word pairs. Rank the pairs in order of decreasing similarity and discuss how close your ranking is to the one obtained experimentally by Miller and Charles. (Note that what matters is the ranking, not the actual scores).

Exercise 3. The main aim of this exercise is to make you reflect about aspects of meaning that only become apparent when we examine the distributional behaviour of some words.

Have a look at this online tool: http://www.scottishcorpus.ac.uk/corpus/bnc/compair.php

(a) What do you think this tool does? You are welcome to search for descriptions of the tool on the web, but explain it in your own words.

(b) The English adverbs *'utterly'* and *'absolutely'* are typically considered synonymous by most thesauri (see e.g. http://thesaurus.com/browse/utterly). Compare them using this tool. What do you observe? What can you conclude about their meaning and usage?

(c) Compare at least one more pair of words and comment on the results.

[1] The mean similarity score for each of the 30 word pairs can be found in the original paper available at http://www.tandfonline.com/doi/pdf/10.1080/01690969108406936.

Exercise 4. This exercise asks you to implement a simple Distributional Semantic Model, to incrementally improve it, and to evaluate its performance informally with respect to how well it captures semantic similarity.

As a starting point, you can use the basic Python implementation of a DSM in the companion file `dsm_basic.py`, which uses some of the capabilities of NLTK. This basic implementation is due to Katrin Erk (see the heading in `dsm_basic.py`). There is also a guide to the code for those of you with little programming experience.

You are welcome to ignore `dsm_basic.py` and instead create your own implementation in your favourite programming language, provided you include clear documentation and sample runs or demos.

The following items assume that you take as a starting point the implementation in `dsm_basic.py`. If you decide to start from scratch, try to go through similar steps (as much as it seems reasonable). Most of the question items below are rather open-ended. They give you pointers on things you may want to try, but you are welcome to make your own decisions depending on your programming skills and your own interests.

You may answer each of the points below in turn or choose to address them in a general report.

(a) Start by briefly describing the characteristics of the basic DSM created with `dsm_basic.py` in terms of the DSM parameters we have seen in class.

(b) As you will quickly realise, the resulting model has several problems. Take, for instance, the similarity scores for a few pairs of words printed at the end when you run the script. Why are they all so high?

(c) Try to improve the model by pre-processing the corpus further before building the vector space. Some possibilities: The NLTK includes a corpus of "stopwords" — a list of high frequency words that have little lexical content (see chapter 2, section 2.4 of the NLTK book) — and several lemmatizers (see chapter 3, section 3.6). Explain how any changes in pre-processing affect the model and give some examples.

(d) Experiment with some other parameters such as the context size or with different corpora. You may also consider more sophisticated steps such as implementing a function to weight the frequency counts, or anything else you think could improve the model. Again, explain how your changes affected the model.

(e) Choose a way to evaluate your model in terms of how well it accounts for semantic similarity. Some possibilities: Consider a few representative examples and compare the performance of your model to that of Web Infomap http://clic.cimec.unitn.it/infomap-query/ or to the word pair rankings experimentally obtained by Miller and Charles (1991), or check whether pairs of words that are part of the same WordNet synset are assigned higher similarity scores by your model than words belonging to different synsets. Feel free to come up with your own evaluation method.