# Distributional semantics practical 1

Aurélie Herbelot

## 1 Requirements

To run the tutorial, you will need the following installed. Instructions are below work on Ubuntu. Ask if you have problems installing the required packages!

- python 2 (you probably have it already, but otherwise check out `https://wiki.python.org/moin/BeginnersGuide/Download`);
- pip:
  sudo apt-get install python-pip python-dev build-essential;
- 'DS tutorial' repository from GitHub. Clone it in whichever directory you want it:
  git clone `https://github.com/minimalparts/Tutorials.git`
- requirements for DISSECT:
  cd Tutorials; pip install -r requirements.txt

## 2 Building your first semantic space

The first thing to do in a count model is to calculate the co-occurrence frequencies between words.

```
#Download a text/corpus:
cd data;
wget http://www.gutenberg.org/ebooks/11.txt.utf-8 -O alice.txt

#Make distributional space with window size +/-2, tagged data.
cd ../utils/;
./mkDSSpace ../data/alice.txt 2
```

```
#See 20 most characteristic contexts
python ./viewdistchars.py Queen_N ../spaces/alice.dm 20|less

#See 20 nearest neighbours
python kneighbours.py ../spaces/alice.pkl Queen_N 20
```

**Exercises:**

- Get a feel for what ends up at the top of the obtained distributions, and what kind of nearest neighbours are returned.
- How do the characteristic contexts and nearest neighbours change if you modify the number of columns and rows in the semantic space? (Try making hypotheses and verifying them by modifying *mkDSSpace*.)
- What changes when you increase the size of the word window?
- What changes when using untagged data?

# 3 Investigating a large semantic space

The spaces/ folder contains a pre-computed space from Wikipedia (PPMI, untagged, dimensionality-reduced to 300 dimensions).

```
cd spaces/
tar -xzvf wikipedia.dm.a.tar.gz
mv wikipedia.dm.a wikipedia.dm
cd ../utils/
python dm2pkl.py ../spaces/wikipedia.dm
```

Try a few nearest neighbours to 'get a feel' for the space:

```
python kneighbours.py ../spaces/wikipedia.pkl queen 20
python kneighbours.py ../spaces/wikipedia.pkl democracy 20
...
```

**Exercises:**

- Read `http://www.aclweb.org/anthology/S12-1012` and become familiar with the clarkeDS and invCL hyponymy measures.

- Try out the hyponymy code:

  ```
  python hyponymy.py ../spaces/wikipedia.dm horse animal
  ```

- Combine the hyponymy and nearest neighbours code to produce a system which returns the likely hypernyms of a word. Your program should be able to take a term and return 3 hyponymys, e.g.:

  ```
  python getHypernyms.py ../spaces/wikipedia.dm cat
  ```

  would ideally return something like *animal, pet, feline*.