Traitement automatique du langage
TP 6 — Distributional semantics

Asheesh Gulati
Exercises prepared by Yves Scherrer & Aurélie Herbelot
22.11.2018

Submit by 05.12.2018 midnight.

Requirements

In this assignment, you will explore the concept of semantic spaces and investigate the notion of word similarity. To do so, you will be using a number of scripts, which you will find in *utils.zip* on Chamilo. These scripts require that you install a number of packages for Python 3 . Execute the following commands in a system terminal (not in the Python environment):

```
$ python -m pip install nltk
$ python -m pip install textblob
$ python -m pip install -U git+https://github.com/sloria/textblob-
    aptagger.git@dev
```

Then, execute the following commands in the Python environment:

```
$ python
>>> import nltk
>>> nltk.download('punkt')
>>> nltk.download('wordnet')
```

As an additional prerequisite, you will need to install the DISSECT project[1]. Instructions to download and set the repository up are available at `http://clic.cimec.unitn.it/composes/toolkit/installation.html`. Make sure to check the "python3" branch out, to install all dependencies (NumPy, SciPy and sparsesvd), and to add the location of the source folder to the PYTHONPATH environment variable. One of the main components of DISSECT enables the creation of semantic spaces from co-occurrence matrices. To learn more about this component and its usage, you can have look at `http://clic.cimec.unitn.it/composes/toolkit/creating.html`.

[1] `http://clic.cimec.unitn.it/composes/toolkit/`

1 Building your first semantic space

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

```
# Create a directory, and unarchive the provided scripts there
# The provided scripts should be under the "utils" subdirectory

# Download a text/corpus:
mkdir data
cd data
wget http://www.gutenberg.org/files/11/11-0.txt -O alice.txt

# Make a distributional space with window size +/-2, tagged data
cd ../utils/
./mkDSSpace ../data/alice.txt 2
# This will create a new subdirectory "spaces"

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

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

Exercises:

1. Describe what kind of words end up at the top of the obtained distributions, and what kind of nearest neighbours are returned.
2. 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*.
3. What changes when you increase the size of the word window?
4. What changes when using untagged data?

2 Investigating a large semantic space

You will find *wikipedia.zip* on Chamilo, which contains a pre-computed space from Wikipedia (PPMI, untagged, dimensionality-reduced to 300 dimensions).

```
# Unarchive the pre-computed space in the "spaces" subdirectory

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:

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

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

3. 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 hypernyms, e.g.:

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

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

Documents to hand in

Please hand in a document containing your responses for exercises 1.1 to 1.4. Also hand in the Python code for exercise 2.3, along with a description of the results achieved.