# 20-00-0546-iv Foundations of Language Technology

Homework 5
Lexical Resources

3. December 2020

Please send Python Notebook files (`.ipynb`) for programming parts and plain text or PDF for essay questions.
In case your submission consists of several files, compress these to a `zip`-file. Indicate clearly which submission corresponds to which question. Include comments in your program code to make it easier readable.
The deadline for the homework is **Thursday, 10.12.2020 17:59 CET**.

## 5.1 Homework

**Homework 5.1** *(4 points) Use the predefined path-based similarity measures* `synset1.path_similarity(synset2)` *to score the similarity of each of the pairs in (b). In case there are several synsets for the two words, take the maximum score over all combinations (for the first pair, there are 10 combinations).*

*(a) Rank the pairs listed in (b) in order of decreasing similarity.*

*(b) How close is your ranking to the order given here, an order that was established experimentally by Miller & Charles (1998):*

- *car-automobile, gem-jewel, journey-voyage, boy-lad, coast-shore, asylum-madhouse, magician-wizard, midday-noon, furnace-stove, food-fruit, bird-cock, bird-crane, implement-tool, brother-monk, lad-brother, crane-implement, journey-car, monk-oracle, cemetery-woodland, food-rooster, coast-hill, forest-graveyard, shore-woodland, monk-slave, coast-forest, lad-wizard, chord-smile, glass-magician, noon-string, rooster-voyage.*

*Spearman rank correlation is widely used to compare two rankings. It goes from* $-1$ *(prefect negative correlation) to* $1$ *(prefect positive correlations) whereas* $0$ *means no correlation. For two lists of distinct elements it can be computed using the formula:*

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

*— where* $d_i$ *is a rank difference of a particular element in the two orderings.*

*Look on the web for more information and compute the Spearman rank correlation using any suitable existing tool (e.g. a spreadsheet or one of the available web tools). Spearman rank correlation is also available in Python as a method in the SciPy library*[1]*:*

```
import scipy
scipy.stats.spearmanr(a, b)
```

*Read the documentation very carefully before using it! Alternatively you can write your own method to compute the correlation using the formula above.*

*Make sure that your result makes sense before submitting. For example, if the ordering that you have produced looks similar to the ordering by Miller & Charles (1998) the correlation value should be positive and close to* $1$.

**Homework 5.2** *(1 point) The main relationship among words in WordNet is synonymy, as between the words "shut" and "close". Synonyms are understood to be words that denote the same concept and are interchangeable in many contexts.*[2]
*Let's put that assumption to the test. Use WordNet to access all synsets of the word "**witch**", as an example, and read their definitions. Are these words easily interchangeable? If not, why?*
*If you wanted to substitute a word with a synonym from that word's synset, how could inappropriate substitutions be avoided? Tip: looking at the context of the word that is to be substituted might be helpful.*

[1] https://www.scipy.org/
[2] https://wordnet.princeton.edu/