

# Semantic Computing

## Tutorial 1

Summer Semester 2018

Link to one Python tutorial: <https://www.learnpython.org/en/>.

### Exercise 1

What is the difference between *Syntax*, *Semantics*, and *Pragmatics*?

### Exercise 2

Let's start with some simple Python operations and tests:

- a) Use Python as calculator and try  $12/(4 + 1)$  and  $3 * 3$ . What happens when you multiply text? Try `['Semantic', 'Computing'] * 5`. As you can see, Python allows you to apply multiplication to lists.
- b) Load the example books that are provided with NLTK by using the command `from nltk.book import *` and call `text2`. How many words are there in the text? How many of these are unique?
- c) Calculate the lexical diversity of the text, that is, the ratio of different unique words to the total number of words.
- d) Try the following two commands `text2.collocations()` and `text6.collocations()`. Take a look at the result. What do you think collocations are?
- e) Consider the following expressions. What is their result and what is the difference between them?
  - `"Semantic Computing"[9:18]`
  - `"Semantic Computing"[9:]`
  - `"Semantic Computing"[-9:]`
  - `["Semantic", "Computing"][1]`
- f) What is the difference to the above and `sent1[2][2]`? `sent1` is the first sentence of the first book in the NLTK books corpus.

### Exercise 3

Implement an algorithm for Word Sense Disambiguation using WordNet. You can use the Lesk algorithm from the lecture slides or implement your own if you prefer. The words to be disambiguated are in file `text.txt` of tutorial 1 of the following format:

Word: target word to be used to query WordNet

SentenceID: used to identify the sentence

Text: Sentence to be used to disambiguate the different senses obtained for the word

In order to evaluate your results, compare the WordNet senses you obtain to the ones in the file `evaluation.txt`. Count the number of correctly identified senses and divide them by the number of

words/sentences provided as input.

Please load the files from the following github: <https://github.com/dgromann/SemanticComputing/tree/master/tutorial1>

#### **Exercise 4**

Take a closer look at the results that you obtained and try to explain what some of the major problems of the incorrect results might have been.

#### **Exercise 5**

The Lesk algorithm is a very basic algorithm that only takes an intersection of words in the definition of the word sense and the input text. What could be done to improve on this basic algorithm? How does this change the results?