# Statistical Natural Language Processing (SS-2018)
# Exercise Sheet 8

**Submission Deadline:** 15.06.2018, 23:59 (CEST)

## Word Sense Disambiguation

1) (1 point) The lower bound of disambiguation accuracy is the performance of the simplest possible algorithm, the baseline system. For the baseline, we usually assign the most frequent sense to all contexts.

   This baseline system's performance depends on how much information is available. Describe at least two diffrent situations in which the baseline system performs very bad and one situation in which it performs very good.

2 (1 point) The two supervised methods, Bayesian Classification (Slide 32) and Information-Theoretic Approach (Slide 37), differ on the number of features used (one vs. many). How would one design a Bayes classifier that uses only one feature and an information-theoretic method that uses many features?

3) (1 point) Is it important to evaluate unsupervised disambiguation on a separate test set or does the unsupervised nature of the method make a distinction between training and test set unnecessary? For example, k-fold cross validation does not make a distinction between test and training set as no separate test set is required

## Lesks Algorithm

Dictionary-based WSD methods rely on the definition of senses in dictionaries and thesauri. In this exercise you will implement a dictionary-based algorithm which compares the context of the word to be disambiguated and the dictionary definitions of the different senses of this word, and selects the most similar sense.

Implement a *simplified* version of the algorithm proposed by Lesk (Slide 27). Here we will treat the contexts and definitions as bags-of-words, and will estimate the overlap between the two *sets* of words using the following similarity meassure:

$$sim_1(X,Y) = \frac{2 \times |X \cap Y|}{|X| + |Y|} \tag{1}$$

The evaluation data[1] for this exercise consist of six sense-annotated English words. Each ambiguous word can have one of two senses (for instance, the word *crane* has the senses *crane%machine* and *crane%bird*). The evaluation data is divided into six files, one per word, and contains text snippets which serve as contexts for disambiguation. Each text snippet is labeled with the correct sense (to be used for evaluation). The definitions of each of the two senses of the word we want to disambiguate are provided in a separate file.

4) (1 point) Discuss the difference between this WSD algorithm and the original algorithm proposed by Lesk (Slide 27). Assume that the both variants use the similatiry meassure $sim_1(X,Y)$.

5) (1 point) Tokenize and normalize the data. Make sure to:

   – remove stop words (using English words from nltk.corpus.stopwords)

   – lowercase

   – remove punctuation (using string.punctuation)

[1] Consult the README provided in the data archive for more information about the structure of the files.

- perform stemming (using nltk.stem.SnowballStemmer)

6) (2 points) Implement a function that returns the most probable sense for a word. The function takes as input the normalized tokens of a text snippet containing the word and the normalized tokens in each of the two sense definitions, and returns the sense with the most overlap according to the similary meassure. In case of ties, select the first sense definition, as the senses have been sorted by their frequency according to WordNet.

7) (1 points) Report the accuracy of your WSD implementation for each of the six words. Compare it to the accuracy of assigning the most frequent sense to every instance.

$$accuracy = \frac{\text{number of correctly identified instances}}{\text{total number of instances}} \tag{2}$$

8) (2 point) How are the results changing if you use the Jaccard similarity coefficient for this task? Again, report the accuracy of your implementation using Jaccards coefficient.

$$sim_{Jac}(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{3}$$

Proove that $sim_1$ and Jaccards coefficient $sim_J$ have a monotonic relationship by prooving that $J = S/(2 - S)$ and $S = 2J/(1 + J)$ where $S$ is $sim_1$ and $J$ is $sim_{Jac}$. Assume that at least one set $X$ or $Y$ is non-empty.

## Submission Instructions

- You must form groups of 2 to 3 people
- Submit only 1 archive file in the ZIP format with name containing the MN of all the team members, e.g.:

  Exercise_06_MatriculationNumber1_MatriculationNumber2.zip
- Provide in the archive:
  i. your code, accompanied with sufficient comments
  ii. a PDF report with answers, solutions, plots and brief instructions on executing your code
  iii. a README file with the group member names, matriculation numbers and emails
  iv. Data necessary to reproduce your results
- The subject of your submission mail **must** contain the string [SNLP] (including the braces) and explicitly denoting that it is an exercise submission, e.g:

  [SNLP] Exercise# Submission MatriculationNumber1 MatriculationNumber2
- Depending on your tutorial group, please send your assignment to the **corresponding tutor**:
  - Mo. 16-18: Lukas Lange *s9lslang@stud.uni-saarland.de*
  - Th. 14-16: Harshita Jhavar *snlp18.thursday@gmail.com*
  - Fr. 10-12: Marimuthu Kalimuthu *mkalimuthu@lsv.uni-saarland.de*
- If two teams submit same solutions, both teams will be given 0 points and no presentation chance will be given to the two team members. If you do this again, you will be disqualified from the exam.