# Dependency Parsing exercises:
# Machine learning

Deadline: 10.05.2021

Please send completed solutions to waszczuk@hhu.de and evang@hhu.de with subject "dependency homework" and attachment "ex4_lastname(s).zip" containing a PDF with the description of your solution and the corresponding Excel sheet.

1. In this exercise, you will design a simple feature representation and manually "train" a linear classifier to solve PP-attachment ambiguities. To keep things simple, you should assume only binary features, and train for a binary prediction whether the prepositional phrase in a sentence attaches to the main verb (1) or to a noun (-1). You can assume UD-style dependencies.

   Consider the following "training corpus":

   | Sentence | outcome |
   |---|---|
   | The astronomer saw the stars with the telescope | 1 |
   | The cat chases the mouse with pointy ears | -1 |
   | The mouse saw the astronomer with the telescope | -1 |
   | The mouse hears the cat with its pointy ears | 1 |

   (a) Design 4 binary features {0,1} that could be helpful in making the prediction, and give a table of the feature values in the training corpus. The value 1 is used when the feature is true for a sentence, 0 when it is not. The features can rely on anything available to a parser, which mostly falls into the following three categories:

      i. Lexical, e.g., the word before the preposition is 'astronomer'
      ii. POS tag, there is an adjective after the preposition.
      iii. Dependency relation/label. You may assume that the dependency heads of the words before the last word (for which the PP-attachement decision is taken) are available, e.g.: there is a dependency from 'mouse' to a verb with the 'sbj' label.

   (b) Enter the table of feature values and the PP-attachment outcomes in an Excel sheet. Add a row for feature weights; the initial values can be arbitrary, e.g. -1 or 3. Add a formula to each row that takes the feature weights and multiplies them with the feature values:

   $$f(x) = \text{sign}(w_1 * f_1 \ldots w_n * f_n)$$

   The Excel sheet should look like this:

   |   | A | B | C | D | E | F | G |
   |---|---|---|---|---|---|---|---|
   | 1 |  | Feature 1 | Feature 2 | Feature 3 | Feature 4 | outcome | Prediction |
   | 2 | Weights | 1.0 | 1.2 | -1 | -1.1 |  |  |
   | 3 | Sentence 1 | 0 | 1 | 1 | 0 | 1 | =sign(B2 * B3 + C2 * C3 ...) |
   | 4 | Sentence 2 | ... |  |  |  | -1 | =sign(B2 * B4 + C2 * C4 ...) |
   | 5 | ... |  |  |  |  |  |  |

The formula in the last column shows you what your classifier would currently predict, and whenever you change the weights, the prediction may change as well.

Adjust the weights manually until the sentences in the training corpus are predicted correctly as much as possible. Start by giving the weight the correct sign: does this feature make it more or less likely that the preposition should be attached to the verb? If more, then the weight should be positive; otherwise negative. The next step is to consider how important the feature is; an important feature should have a larger weight.

(c) Now apply this model to a new, unseen sentence:

| Sentence | outcome |
| --- | --- |
| The cat chases the mouse with the astronomer | 1 |

Extract the features from this sentence and apply the formula to generate a prediction. Does your model make the right prediction? You can of course cheat and adjust the weights until this sentence is also correctly predicted, but note that this is a big methodological *faux pas*, "training on the test set."