DD1418 Language Engineering with Introduction to Machine Learning
Johan Boye
2019-10-22

Assignment 1

Readings: Read section 2.5 in Jurafsky-Martin.

Code: The skeleton code can be downloaded from Canvas or from

`http://www.csc.kth.se/~jboye/teaching/language_engineering/a01/Aligner.zip`

Unzip the code in your home directory. Go to the folder `Aligner` and type:

```
pip install -r requirements.txt
```

Now everything needed for the assignment should be installed.

Problems:

1. Minimum-cost string alignment is an important task in many NLP applications, as well as in bioinformatics. The `Aligner.py` file contains a skeleton Python program for computing and printing the minimum-cost alignment of two strings or two files. **Your task is to extend the code so that the program works correctly** (look for the comments `YOUR CODE HERE` in the program). Use the scripts `run_aligner_01.sh` to `run_aligner_08.sh` to run the program on various test examples, and you can also invent your own test examples if you like. The `check` flag will make the program compare your result with the correct alignment (the alignment resulting from the method presented in the lecture). E.g. :

   ```
   python Aligner.py -s broke above --check
   ```

   However, note that, in general, there are several equally good alignments. For instance, when aligning *step* with *steep*, the 'e' in *step* can be aligned with either the first or the second 'e' in *steep*.