Counting in Python

Guillaume Wisniewski
wisniews@limsi.fr

February 2018

The goal of this lab is to get familiar with the python programming language.

1 Data

In this lab, we will consider several corpora of the Universal Dependency Project. The UD project aims at providing grammatical annotations for a wide array of languages (it contains corpora for more than 70 languages in its latest release). The files we will be using can be downloaded from the lecture web site.

Figure 1 shows an example of the `conllu` format used in the UD project. In the rest of this lab, we will use the following characteristics of this format:

- blank lines mark sentence boundaries;
- lines starting with a sharp have to be ignored;
- each line describes one word of the sentence; information about this word are stored in tab-separated columns;
- lines in which the first column contains an hyphen should be ignored;
- the 4-th column contains the Part-of-Speech label of the word: the word is a verb if its PoS is either `VERB` or `AUX`;
- the 7-th column can be used to identify verbs and passive constructions: if its value *contains* `"nsubj"`, there is a verb in the sentence, if its value *is* `"nsubj:pass"`, the construction is in the passive voice.

2 Appetizer

1. Write a function that returns the number of words in the GSD corpus (i.e. the number of *tokens*).

```
# sent_id = fr-ud-dev_00001
# text = Aviator, un film sur la vie de Hughes.
1       Aviator Aviator PROPN   _       _       0       root    _       SpaceAfter=No
2       ,       ,       PUNCT   _       _       1       punct   _       _
3       un      un      DET     _       Definite=Ind|Gender=Masc|Number=Sing|PronType=Art       4       det     _       _
4       film    film    NOUN    _       Gender=Masc|Number=Sing 1       appos   _       _
5       sur     sur     ADP     _       _       7       case    _       _
6       la      le      DET     _       Definite=Def|Gender=Fem|Number=Sing|PronType=Art        7       det     _       _
7       vie     vie     NOUN    _       Gender=Fem|Number=Sing  4       nmod    _       _
8       de      de      ADP     _       _       9       case    _       _
9       Hughes  Hughes  PROPN   _       _       7       nmod    _       SpaceAfter=No
10      .       .       PUNCT   _       _       1       punct   _       _

# sent_id = fr-ud-dev_00002
# text = Les études durent six ans mais leur contenu diffère donc selon les Facultés.
1       Les     le      DET     _       Definite=Def|Gender=Fem|Number=Plur|PronType=Art        2       det     _       _
2       études  étude   NOUN    _       Gender=Fem|Number=Plur  3       nsubj   _       _
3       durent  durer   VERB    _       Mood=Ind|Number=Plur|Person=3|Tense=Pres|VerbForm=Fin   0       root    _       _
4       six     six     NUM     _       _       5       nummod  _       _
```

Figure 1: Example of `conllu` format.

2. Write a function that returns the number of *unique* words (i.e. the number of types[1])

Advice: when developing / testing your function, consider a small corpus made of one or two sentences so that you can manually work out the expected answer.

3 Probability Estimation

This Section aims at estimating the probability that a French sentence is written in the passive voice.

3. Write a function that returns a list of all the sentences contained in a `conllu` file.

4. Write a function that returns the number of verbs in a sentence *and* the number of verb in the passive form

5. Compute the ratio "number of passive constructions over number of verbs" for 5, 50, 100, 1,000, 5,000, 10,000, 20,000 and 50,000 sentences. What can you conclude?

6. Explain the code of Figure 2. In particular:
   - what is happening line 12? what is the type of `content`?
   - what is line 14 doing? Why are we using square brackets and not parentheses like line 12?
   - Explain line 17 and 19.

[1] In the sentence `"a cat and a dog"` there are 4 types (`a`, `cat`, `dog` and `and`) and 5 tokens?

```python
from collections import Counter
from random import sample
from pathlib import Path


UD_PREFIX = Path().home() / Path("workspace/corpus/ud-treebanks-v2.3")
filename = Path(UD_PREFIX, "UD_French-GSD", "fr_gsd-ud-train.conllu")


with open(filename, "r") as ifile:

    content = (line.split("\t") for line in ifile
               if line.strip() and not line.startswith("#"))
    dependencies = [line[7] for line in content if "-" not in line[0]]

    for n_sentence in [100, 500, 1000, 5000, 10000, 20000, 30000, 40000]:
        c = Counter(sample(dependencies, n_sentence))

        n_verb = sum(v for k, v in c.items() if "nsubj" in k)
        n_pass = c["nsubj:pass"]

        print(n_pass / n_verb)
```

Figure 2: Code to analyse.