Week 1 : POS tagging warm-up – With your hands

During the whole course, we will apply the techniques seen during lectures to a single archetypal yet central NLP task : POS tagging. We will go from rule based systems to statistical ones, to machine learning techniques to end up with the most up-to-date deep learning based language models.

The reason for sticking to POS tagging is that despite its apparent simplicity, it is a very complete task that can be solved with an array of techniques, and that most of if not all other NLP task can be seen in a way of another as either a tagging tasks, a structure prediction task or a sequence prediction task.

This week, we are starting easy, just loading the data and try a few hand made rules to see how far we can get with it.

---

1. Get the data : Go on the Universal Dependencies website and download the French GSD, English EWT, Finnish TDT. (If you want you can download the whole UD 2.12, it may be a bit long though.)
2. Look at the data. You should always know the kind of data you're working with. Understand its format, its structure, its meaning, the available information.
3. Make a CoNLL-U parser. Just keep lines starting with an integer and no dash or dot before the first tabulation. We only need the second and fourth columns for each word in each sentence.
4. Look at the data (again), but now with Python code. Count tokens, sentences, length distributions, POS tags, average number of POS tags per tokens... You should not cheat and learn the test set by heart, but again, you should know the data you're working with. This is important to choose the good methods and to detect potential bugs.
5. Start to write rules for French and/or English based on the training set, test them on both the training set and the dev set. Begin with simple rules that only look at the word in question. Do not do learning, stick to basic hand written hard coded rules like "if x ends in y then POS tag is z". Try one or two rules per major tags. Do not write more then 30 rules.
6. Now, write rules using also the left and right neighbours as feature sources. How high do you get?