PoS Tagging with a Perceptron

Guillaume Wisniewski
guillaume.wisniewski@u-paris.fr

September 2021

The goal of this lab is to implement a tagger that relies on the perceptron algorithm to predict the PoS of the words of a sentence and evaluate it on the corpora of the UD project. This lab is made of 4 parts that correspond to the usual steps of a ML project:

1. reading the corpora;
2. features extraction;
3. parameter estimation;
4. evaluation.

In all our experiments we will consider the French GSD corpus.

1 Corpus reading

- Write a function that takes a filename as parameter and returns a list of examples. Each example is made of pairs (observation, label) in which `observation` is a sentence (a sequence of words) and `label` is a sequence of labels (there is one label for each word).
- Plot the distribution of labels in the train and test corpus. How many examples are there in the train set ? in the test set ?

A description of `conllu` format can be found at https://universaldependencies.org/format.html. When reading data, it is necessary to pay attention to multword tokens (indexed with integer ranges)

2 Feature extraction

We will consider X simple feature templates to describe the i-th word of a sentence $\mathbf{w} = w_1, ..., w_0$:

- the current word w_i;
- the previous word w_{i-1};
- the following word w_{i+1};
- the word w_{i+2};
- the word w_{i-2};
- a biais (i.e. a feature that is always present);
- a binary feature that is true when the word starts with a capital letter;
- a binary feature that is true when the word contain at least one number.

As usual the feature vector will be represented by a sparse vector and each word will be described by a list of strings corresponding to the non-zero features. For instance, the list of features describing the 3rd word of the sentence "`Mélina et Mélio dorment dans leur chambre.`" is: `"curr_word_Mélio"`, `"prev_word_et"`, `"prev_prev_words_Mélina"`, `"next_word_dorment"` `"next_next_word_dans"`, `"biais"`, `"starts_with_upper"`

1. Write a function that takes a corpus (list of sentences and their label) as input and return a list of pairs (feature vector, label).
2. What is a dimension of feature vector? How many non-zero features are there in general?

3 Averaged perceptron implementation

We will now implement the averaged perceptron algorithm to