



The Python extension package named `nltk`<sup>1</sup> provides a set of tools that are useful for addressing information extraction problems such as Named Entity Recognition (NER). More specifically, you can use the following methods:

- `nltk.sent_tokenize(d)`, which splits a document `d` into a list of sentences;
- `nltk.word_tokenize(s)`, which splits a sentence `s` into a list of words;
- `nltk.pos_tag(w)`, which leverages a sequence classification model to tag the words in list `w` according to their part-of-speech (i.e., tag words according to morphosyntactic classes such as noun, verb, adjective, ...);
- `nltk.ne_chunk(p, binary=True)`, which tags the words in list `p` as named entities or not (where each word in `p` was previously tagged with a part-of-speech tag).

Note that the output of each of these tools can be used as input to the next tool.

The `nltk`<sup>2</sup> documentation also presents several alternative models for parts-of-speech tagging and named entity recognition, leveraging different types of algorithms (e.g., structured perceptrons, CRFs, etc.)

## 1

Test the Senna<sup>3</sup> POS tagger, NER Tagger and Chunk Tagger with a few sentences of your own, or extracted from Web sites. Try text from different contexts (e.g. news, blogs, etc.).

## 2

Using the above tools, print all named entities found in the documents of the 20 newsgroups collection<sup>4</sup>. This document collection can be conveniently accessed through the scikit-learn library, as shown in the previous lab class.

---

<sup>1</sup><http://www.nltk.org>

<sup>2</sup><https://www.nltk.org/api/nltk.tag.html>

<sup>3</sup><https://www.nltk.org/api/nltk.tag.html#module-nltk.tag.senna>

<sup>4</sup> <http://qwone.com/~jason/20Newsgroups/>

### 3 Pen and Paper Exercise

Consider the Hidden Markov Model from the previous exercises, represented by the following probabilities. Remember that  $\pi$  corresponds to the initial probabilities of each state,  $B$  corresponds to the state emission probabilities, and  $A$  corresponds to the transition probabilities.

The symbols corresponding to each line in matrix  $B$  are  $a$ ,  $b$ , and  $c$ .

$$\pi = (0.8 \quad 0.2) \quad B = \begin{pmatrix} 0.1 & 0.6 \\ 0.7 & 0.2 \\ 0.2 & 0.2 \end{pmatrix} \quad A = \begin{pmatrix} 0.1 & 0.5 \\ 0.9 & 0.5 \end{pmatrix}$$

Consider now a structured perceptron in which the considered feature representations/scores correspond to restructuring the HMM probabilities as scores.

- (a) Consider the observation **acb** associated to the sequence of states **121**. Show how this observation would be represented in terms of binary features.
- (b) Consider a structured perceptron defined with feature weights corresponding to the logarithm of the HMM probabilities. What is the most likely sequence of states for the sequence of symbols **acbc**?
- (c) Starting from the structured perceptron model from the previous question, compute a new model using one iteration of the structured perceptron training method, assuming that you had only one observation available: **acb** associated to the sequence of states **121**.