Traitement automatique du langage
TP 7 — Identify gene names with HMMs and the Perceptron algorithm

Haozhou Wang
Exercises prepared by Asheesh Gulati
05.12.2019

Submit by 19.12.2019 midnight.

In this assignment, you will build a trigram hidden Markov model to identify gene names in biological text. However, instead of using maximum likelihood parameter estimates, as you did in TP3 (Part 2, A trigram HMM tagger), you will this time use the estimation algorithm described in section 2.1 of the following paper by Michael Collins: `http://www.aclweb.org/anthology/W02-1001.pdf`.

Perceptron

The perceptron algorithm was originally used for training binary classifiers. You can have a look at `https://nbviewer.jupyter.org/github/Christof93/perceptron/blob/master/perceptron_algorithm.ipynb` to catch the gist of the underlying model. In the context of this assignment, you are expected to implement the variant of the perceptron algorithm described in the paper by Michael Collins. The idea is to consider that the emission and transition parameters are weights in a single-layer neural network. When learning the model, these weights are updated upon reading, in turn, each tagged sentence in the training set.

Compare the results obtained using this approach with the results obtained in TP3. How many iterations are necessary to reach acceptable results?

Documents to hand in

Please hand in the Python code for for your tagger, along with a description of the results achieved.

Algorithm 1 Perceptron Algorithm for POS Tagging

Require: training set containing S sentences
  initialize $\alpha_{x,y,z} \leftarrow \mathbf{0}$ and $\alpha_{t,w} \leftarrow \mathbf{0}$
  repeat
    for $i = 1 \dots S$ **do**
      $z_{[1:n_s]}$ = best tagged sequence {using Viterbi decoder}
      for every tag trigram $\langle x, y, z \rangle$ **do**
        $c_1 \leftarrow$ count of $\langle x, y, z \rangle$ in $t^i_{[1:n_i]}$
        $c_2 \leftarrow$ count of $\langle x, y, z \rangle$ in $z_{[1:n_i]}$
        if $c_1 \neq c_2$ **then**
          $\alpha_{x,y,z} \leftarrow \alpha_{x,y,z} + c_1 - c_2$
        end if
      end for
      for every tag/word pair $\langle t, w \rangle$ **do**
        $c_1 \leftarrow$ count of $\langle t, w \rangle$ in $\left(w^i_{[1:n_i]}, t^i_{[1:n_i]}\right)$
        $c_2 \leftarrow$ count of $\langle t, w \rangle$ in $\left(w^i_{[1:n_i]}, z_{[1:n_i]}\right)$
        if $c_1 \neq c_2$ **then**
          $\alpha_{t,w} \leftarrow \alpha_{t,w} + c_1 - c_2$
        end if
      end for
    end for
    shuffle training set
  until convergence or maximum number of iteration