Exercise Sheet 4

Due Date: January 9, 10 pm

Note on Submission

All solutions have to be uploaded together as a single zip file to LernraumPlus. Solve the exercises by implementing the functions in the file `exercise_sheet4.py`.

In this exercise sheet, you will create your own linear chain conditional random field by implementing the methods of the class `LinearChainCRF` in the file `exercise_sheet4.py`. We use the corpus and the feature types we already used in the previous exercise sheet:

$$f_{w/t}(x_i, y_i) = \begin{cases} 1 & \text{iff } x_i = w \quad \wedge \quad y_i = t \\ 0 & \text{otherwise} \end{cases}$$

$$f_{t_1/t_2}(x_i, y_i) = \begin{cases} 1 & \text{iff } y_{i-1} = t_1 \quad \wedge \quad y_i = t_2 \\ 0 & \text{otherwise}\,, \end{cases}$$

where w is a word, t, t_1, t_2 are labels, x_i is the word of a sentence at position i, y_i is a label we assigned to this word and y_{i-1} is the label of the word at position $i-1$. The factors $\Psi_t(y_t, y_{t-1}, x_t)$ we use have the following form:

$$\Psi_t(y_t, y_{t-1}, x_t) = \exp\left(\sum_{i=1}^{F} f_i(y_t, y_{t-1}, x_t) \cdot \theta_i\right) ,$$

where F is the number of all features, f_i is the indicator function for the i-th feature and θ_i is the weight assigned to the i-th feature. You may use the code you developed in the previous exercise sheet to handle the features (building the set of features which are observable in the given corpus, estimating the active features, ...). Hint: Use sets instead of vectors to store the active features in order to improve performance. Don't forget to introduce the special label `start` to model the previous label of the first word of a sentence.

Exercise 1 – Training a Conditional Random Field [2+2+2+2+2 points]

a) Implement the methods `forward_variables` and `backward_variables` to compute the forward variables $\alpha_t(j)$ and backward variables $\beta_t(i)$, respectively, for a given sentence $\vec{x} = (x_1, \ldots, x_T)$.

b) Implement the method `compute_z` to compute the partition function $Z(\vec{x}) = \sum_{y_i,\ldots,y_T} \prod_{t=1}^{T} \Psi_t(y_t, y_{t-1}, x_t)$ which is needed for normalization. Use the forward or backward variables.

c) Implement the forward-backward algorithm to compute the marginal probability $p(y_t, y_{t-1}|\vec{x})$ for a given sentence $\vec{x}$. Put your code into the method `marginal_probability`.

d) Implement the method `expected_feature_count` to compute the expected feature count $E^k_{\vec{\theta}}(\vec{x}) = \sum_{t=1}^{T} \sum_{y,y'} f_k(y, y', \vec{x}) p(y_t = y, y_{t-1} = y'|\vec{x})$ for the k-th feature given a labeled sentence $\vec{s} = ((x_1, y_1), \ldots, (x_T, y_T))$.

e) Implement the method `train` to train the conditional random field via gradient ascent. Recall that the partial derivate of the log likelihood function (for a labeled sentence $\vec{s}$) with respect to parameter θ_k is given by

$$\frac{\partial \log(p(\vec{s}|\vec{\theta}))}{\partial \theta_k} = E^k(\vec{s}) - E^k_{\vec{\theta}}(\vec{s}) \,,$$

where $E^k(\vec{s}) = \sum_{t=1}^{T} f_k(y_t, y_{t-1}, x_t)$ is the empirical feature count of the k-th feature for a labeled sentence $\vec{s}$. Randomly select one sentence from the corpus in each training iteration. Think about experimenting with different learning rates.

Hint: Precompute the empirical feature count for each sentence in the corpus.

Exercise 2 – Maximum-A-Posteriori Label Sequence [10 points]

Implement the Viterbi algorithm for linear chain conditional random fields to compute the most likely sequence of labels given a sentence $\vec{x}$. Put your code into the method `most_likely_label_sequence`.