Exercise Sheet

Statistical Natural Language Processing

Exercise Sheet 3

Due Date: December 12, 8 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_sheet3.py`.

In this exercise sheet, you will create your own maximum entropy model by implementing the methods of the class `MaxEntModel` in the file `exercise_sheet3.py`. We use features of the following form (we already used them in the example discussed in the lecture):

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

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

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 $i-1$. That is, $f_{w/t}(x_i, y_i)$ is 1 if and only if the i-th word of a given sentence is w and if we assign the label t to this word. Similarly, $f_{t_1/t_2}(x_i, y_i)$ is 1 if and only if the label of the $(i-1)$-word of a given sentence is t_1 and if we assign the label t_2 to the i-th word.

Use the provided corpus `corpus_pos.txt` for training your model. You can use the Python function `import_corpus` for importing this corpus.

Exercise 1 – Building the Feature Set [3+2 points]

First of all, you need to build the set $\mathcal{F}$ of all features. Therefore, do the following things:

a) Build the set $\mathcal{X}$ of all words and the set $\mathcal{Y}$ of all labels occurring in the corpus. Use the variable `labels` to save the set of all labels. Then build the set of all features

$$\mathcal{F} = \{f_{w/t} | w \in \mathcal{X}, t \in \mathcal{Y}\} \cup \{f_{t_1/t_2} | t_1, t_2 \in \mathcal{Y}\}\,.$$

You can use pairs of strings to represent the features $f_{w/t}$ and f_{t_1/t_2}. Note that there is no previous word for the first word of a sentence. In this case, the label of the previous word is `start`.

For example, assume we are given the ``sentence'' ((the, DT), (dog, NN)) and we decided to represent features as pairs of strings, then we would build the following sets:

$$\begin{aligned}
\mathcal{X} &= \{\texttt{the}, \texttt{dog}\} \\
\mathcal{Y} &= \{\texttt{start}, \texttt{DT}, \texttt{NN}\} \\
\mathcal{F} &= \left\{ \begin{array}{l}
(\texttt{the}, \texttt{start}), (\texttt{the}, \texttt{DT})(\texttt{the}, \texttt{NN}), \\
(\texttt{dog}, \texttt{start}), (\texttt{dog}, \texttt{DT}), (\texttt{dog}, \texttt{NN}), \\
(\texttt{start}, \texttt{start}), (\texttt{start}, \texttt{DT}), (\texttt{start}, \texttt{NN}), \\
(\texttt{DT}, \texttt{start}), (\texttt{DT}, \texttt{DT}), (\texttt{DT}, \texttt{NN}), \\
(\texttt{NN}, \texttt{start}), (\texttt{NN}, \texttt{DT}), (\texttt{NN}, \texttt{NN})
\end{array} \right\}
\end{aligned}$$

Then, assign each feature to a unique index $i \in \{1, \ldots, |\mathcal{F}|\}$. Then the feature assigned to index i can be referenced by f_i. Finally, initialize the vector θ of parameters. Do all this inside the method `initialize`.

b) Implement the method `get_active_features` which returns a vector $\vec{f} \in \{0,1\}^{|\mathcal{F}|}$ indicating which features are active for a given word x_j. I.e., for a given word x_j and a label y_j for this word, the i-th element of this vector is set to 1 if and only if $f_i(x_j, y_j) = 1$ and 0 otherwise.

Hint: Use a Python dictionary to store the features and their indices.

Exercise 2 – Computing Conditional Probabilities [2+2 points]

a) Implement the method `cond_normalization_factor`, which returns the normalization factor $1/Z(x_i)$, where

$$Z(x_i) = \sum_{y' \in \mathcal{Y}} e^{\vec{\theta} \cdot \vec{f}(x_i, y')} .$$

Note that the label y_{i-i} of the previous word is implicitly given by the index i, but you have to provide it explicitly in the code.

b) Implement the method `conditional_probability`, which computes the conditional probability

$$P(y|x_i) = \frac{1}{Z(x_i)} e^{\vec{\theta} \cdot \vec{f}(x_i, y)} .$$

of a label y given the word x_i.

Exercise 3 Empirical and Expected Feature Count [1 + 3 points]

Implement the method `empirical_feature_count`, which returns the vector $E[\vec{f}(x_i, y_i)]$ of the empirical feature count, and the method `expected_feature_count`, which returns the vector $E_{\vec{\theta}}[\vec{f}(x_i)]$ of the expected feature count, given the parameters $\vec{\theta}$ of the current model.

Exercise 4 Training the Model [2+3+2 points]

a) Implement the method `parameter_update`, which performs one learning step according to

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \alpha(E[\vec{f}(x_i, y_i)] - E_{\vec{\theta}}[\vec{f}(x_i)]) ,$$

where α is a learning rate.

b) Implement the method `train` for training your model via gradient descent. Use the method `parameter_update` you implemented above.

c) Implement the method `predict`, which predicts the most probable label of a given word x_i at position i in a sentence.