BBM 497: Introduction to NLP Lab.

ASSIGNMENT II

Handed out: 24.03.2017 *Handed in: 14.04.2017*

Please submit your solution (code and a PDF of your report) by 17:00pm on the due date. Please describe your code in a separate report. Your reports should not exceed a page.

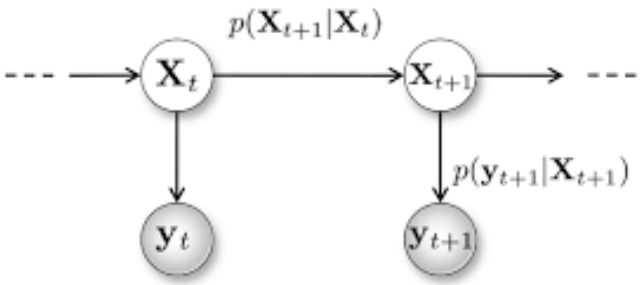
Code and data

All files that are necessary to do the assignment are contained in a zip file which you can get from Piazza.

1 Hidden Markov Models and Part-of-Speech Tagging

In this assignment, you will use Brown corpus which has got all the words tagged with a PoS tag. You will use this corpus in order to train a hidden Markov model for PoS tagging. Therefore, for any given sentence, your program will be able to find the PoS tag of each word.

Task 1. You will **load the entire corpus** by taking each line as a sentence with its PoS tags. Brown corpus consists of several files. You have to read all of them. Each sentence is in the form of "word/pos tag" (fire/nn means that the word 'fire' has the tag 'nn' –which is *noun, the/at* means that the word 'the' has the tag 'at' –which is article).



Task 2. You will **build your hidden Markov model** by initializing the internal variables:

1. The initial tag probabilities $p(t_i)$: the probability that a sentence begins with tag t_i
2. The transition probabilities $p(t_{i+1} \mid t_i)$: the probability that tag t_{i+1} is seen after the tag t_i
3. The emission probabilities $p(w_i \mid t_i)$: the probability that token w_i is generated by tag t_i.

Task 3. Your program will **assign the most probable tags** for the input tokens taken from the input_tokens.txt. Remember that the assignments are based on $\arg\max_i p(w_i \mid t_i)$.

input_tokens.txt file will include only a sequence of words which are all seen in the training corpus. A sample file format is given below:

```
The detached house is far from here .
I almost ran over the snake .
```

Your output will be in the given format:

```
The/at detached/jj house/nn is/bez far/rb from/in here/rb ./.
I/ppss almost/rb ran/vbd over/in the/at snake/nn ./.
```

Task 4. You will **implement the Viterbi algorithm** in this task. Your Viterbi method will find the path with the highest probability by looking at all the possible tag sequences. Your Viterbi algorithm will consists of two steps:

1. You will compute the probability of the most likely tag sequence.
2. You will trace the back pointers to find the most likely tag sequence from the end to the beginning.

Input and output format will be the same as the input/output file format given in the previous task (Task 3). The name of your input file will be test_set.txt.

Submission

You need to implement either in Java or Python. Please submit your source codes and a one-page report in the following submission format.

Submit Format:

This file hierarchy must be zipped before

→ <student id>
→ code.zip
→ report.pdf