[COM4513-6513] Lab 4: Viterbi and Beam search

Instructor: Nikos Aletras

Teaching Assistants: George Chrysostomou, Hardy and Zeerak Waseem

The goal of this lab session is to accelerate the structured perceptron using two methods, Viterbi and Beam search. You should use the same data as in Lab 3.

- [**2 marks**] Implement the Viterbi algorithm to accelerate the argmax operation during both training and testing. Recall that the Viterbi is an exact search algorithm, i.e. it should return exactly the same result as the naive implementation, only faster. To ensure this happens, pay attention to random seeds, use of (ordered) dictionaries, etc. Here is an adaptation of Viterbi for the structured perceptron:

: word sequence $\mathbf{x} = [x_1, ..., x_N]$, weights $\mathbf{w}$

set matrix $V^{|\mathcal{Y}| \times N} = 0$ set backpointer

$$B^{|\mathcal{Y}| \times N} = 0$$
$$V[y, n] = \max_{y' \in \mathcal{Y}} V[y', n-1] + \mathbf{w} \cdot \phi(y, y', \mathbf{x})$$
$$B[y, n] = argmax_{y' \in \mathcal{Y}} V[y', n-1] + \mathbf{w} \cdot \phi(y, y', \mathbf{x})$$

- [**2 marks**] What speed up do you get with Viterbi compared to the standard structured perceptron?
- [**2 marks**] Implement beam search to accelerate the argmax. Recall that it is an inexact search method so the results might differ. But given a large enough beam size it should achieve the same results as viterbi and the structured perceptron from Lab 3.
- [**2 marks**] Does beam search affect your accuracy? What is the effect of beam size on speed and accuracy compared to the standard structured perceptron and Viterbi? **Tip 1:** Try three different beam sizes and report the results. **Tip 2:** Use a table to summarise the results (accuracy and training time) for the standard perceptron (from the lab 3), Viterbi and Beam search.

Submission Details

You should submit a python file (lab4.py) that can be executed as:

- `python3 lab4.py -v train.txt test.txt` for Viterbi
- `python3 lab4.py -b train.txt test.txt` for Beam search

and returns the micro F1 for the phi_1 feature set from Lab 3. Use python's `random` library to fix the random seed so that your results are reproducible! Make sure your code is **Python 3** compatible. You can use argparse for the command line arguments `-v` and `-b`.

You also need to accompany the code with a lab4.pdf (no more than two A4 pages) answering the questions above.

This lab will be marked out of 8. It is worth 8% of your final grade in the module.

The deadline for this assignment is the beginning of the Week 9 lab and it needs to be submitted via MOLE. Standard departmental penalties for lateness will be applied.