Assignment 2

Wang Entang

November 13, 2024

1 Instructions

To run the tagger, you should install the dependencies listed in README and run the main function of the corpus_handler.py file. If you want to change the increment of data size, please modify the Variable size_increment in line 187 corpus_handler.py. A bigger increment number will save running time but reduce the plot quality. To test if the result is right, just run the test code uncommented below the file corpus_handler.py. I also provide the example code in viterbi.py 's main function to test if the viterbi algorithm is right.

2 Extra: plot a learning curve

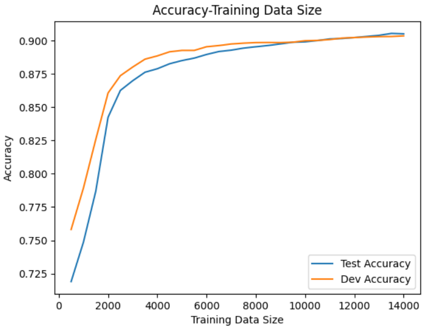

Figure 1: learning curve

2.1 Observations

The x-axis represents the training data size (sentences number), and the y-axis represents the accuracy of the tagger. There are two lines in this plot: Test Accuracy (in blue): represents the accuracy of the tagger on the test dataset. Dev Accuracy (in orange): represents the accuracy of the tagger on the development dataset. Both curves show a sharp increase in accuracy as the training data size increases, especially in the range from 0 to around 4000 data points. After around 4000 training samples, the accuracy gains start to slow down, and both curves begin to plateau. Beyond around 10,000 samples, the accuracy of both the test and development sets becomes relatively stable.

2.2 Conclusions

The tagger shows strong improvement with increased data, especially early on. However, after a certain point (around 4000 samples), the improvements become much smaller. This indicates diminishing returns from adding more data, as the model's performance starts to stabilize. The model reaches its peak performance with around 10,000 training samples. Beyond this point, adding more training data does not lead to significant accuracy improvements.

3 Extra: plot a speed vs. sentence length curve

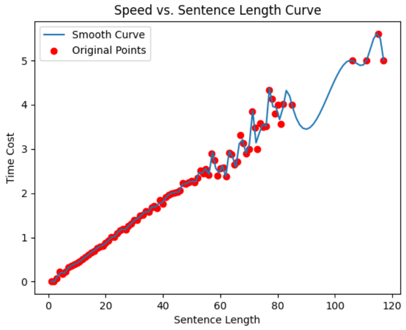

Figure 2: speed vs. sentence length curve

3.1 Observations

The x-axis represents the sentence length (number of words), and the y-axis represents the time cost to process each sentence (unit: 0.001s). The plot contains two components: Smooth Curve (in blue): A cubic spline interpolation that provides a smooth trend line. Original Points (in red): The actual time cost for each sentence length as individual data points. There is a clear positive correlation between sentence length and processing time, as the time cost generally increases with sentence length. The relationship appears to be almost linear for shorter sentence lengths, but as the sentence length grows (above around 80 words), there is more variability in the time cost, and the trend becomes less predictable. I speculate it's because there are less long length samples so the randomness of time cost is big in long length sentences.

3.2 Conclusions

Linear Scaling with Short Sentences: For shorter sentences, processing time scales roughly linearly with sentence length. This suggests that the tagger's complexity grows linearly with sentence length under typical conditions. Increased Variability with Longer Sentences: For longer sentences, the processing time becomes less consistent. This could be due to more complex sentence structures in longer sentences, which might require additional processing or lead to variations in computational efficiency. Efficiency Consideration: The tagger generally performs efficiently for typical sentence lengths (up to around 80 words). For sentences longer than this, the variability suggests that further optimization may be needed for handling very long sentences consistently.

References