DL Assignment 4 Bonus

Meinan Gou meinang@kth.se

1 Introduction

I implemented and compared two RNN networks with different sequence length, one is 25 and another one is 50. I plotted the loss curves and saved the generated tweets from the two networks.

2 Functions

Compared to the mandatory part, I added one more class **Rewrite** to rewrite the tweets in a separate file. I also made some modifications in class **Readfile** due to the big amount of data. I used the first 1,100,000 chars for training in both networks.

3 Results

3.1 Sequence length=25

Figure 1 and 2 shows the loss and synthesized text using sequence length=25. At iter0, the text contains a lot of emojis that cannot be shown at the terminal. As the number of iteration increases, the synthesized text starts to make some sense. Words like "Game of Thrones" and something like a link appears. The synthesized text resembles real tweets as the number of iteration increases.

3.2 Sequence length=40

As is shown in Figure 3 and 4, the evolution of the synthesized text has similar trend as in seq length=25. As the number of iterations increases, the synthesized text contains more and more meaningful words.

4 Conclusions

Compared with Figure 1, Figure 3 has higher value of loss. This is because larger value of sequence contains more samples in one batch, which increases the probability of misclassification. I compared the final result of the two networks, and I found that the result produced by seq length=25 is better than seq length=40 with the same epochs (8) of training. I think the there are mainly two reasons contributing to it. The first one is if the seq length is large, the characters at the end of the text lack of chance to train. The second one is that in SGD, the network would focus on the overall loss of the batch. Common words ("the", "a") would have larger chance to train. If the seq length is long, some uncommon words may not be well trained, which would negatively affect the performance.

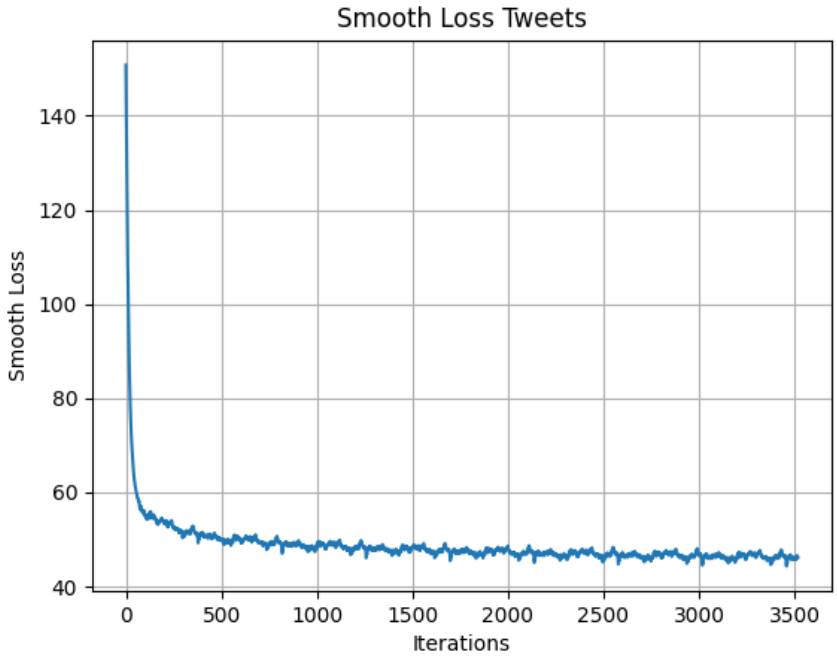
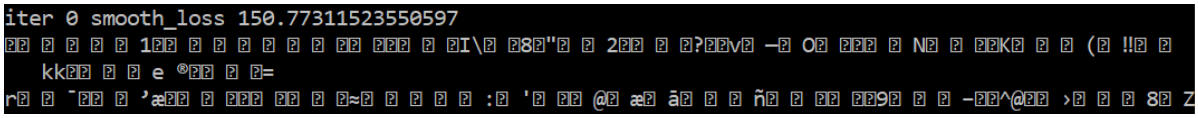
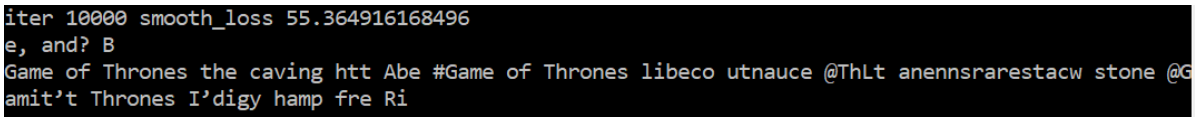


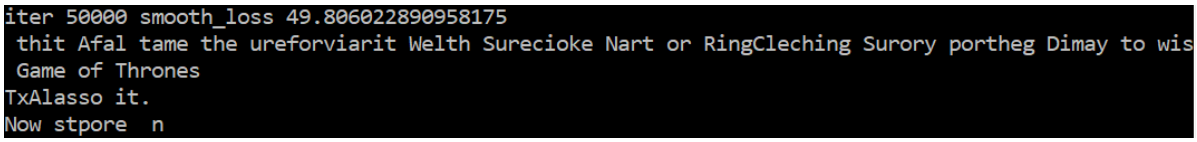
Figure 1: seq=25 Loss



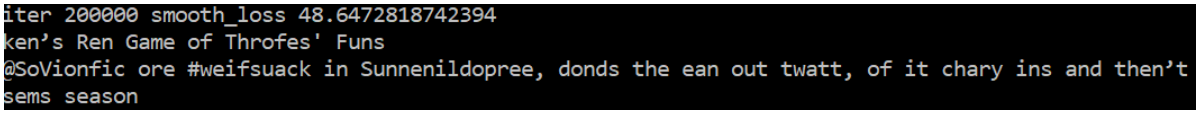
(a) iter0



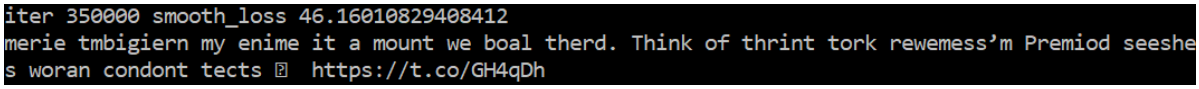
(b) iter10000



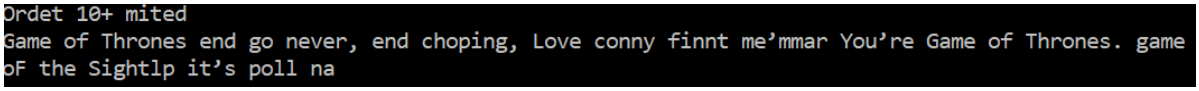
(c) iter50000



(d) iter200000



(e) iter350000



(f) final

Figure 2: Synthesized Text Seq=25

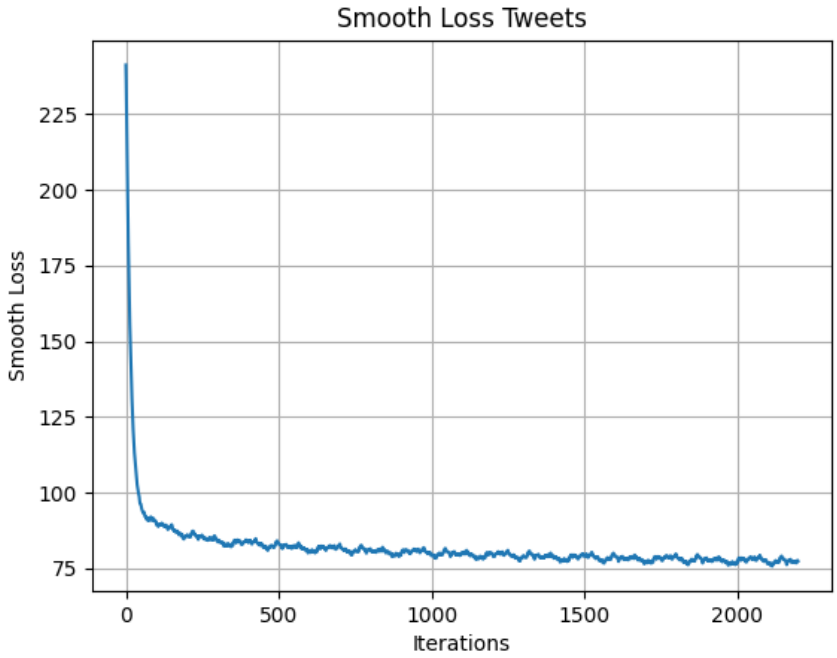
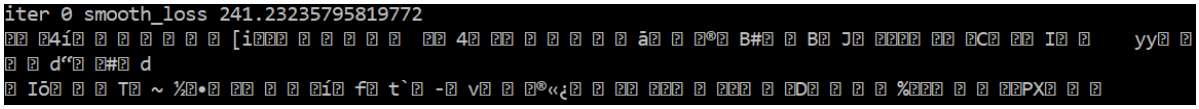
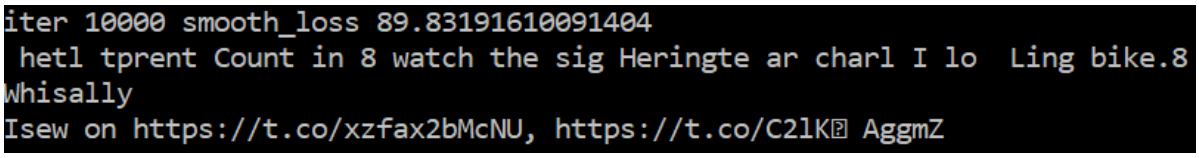


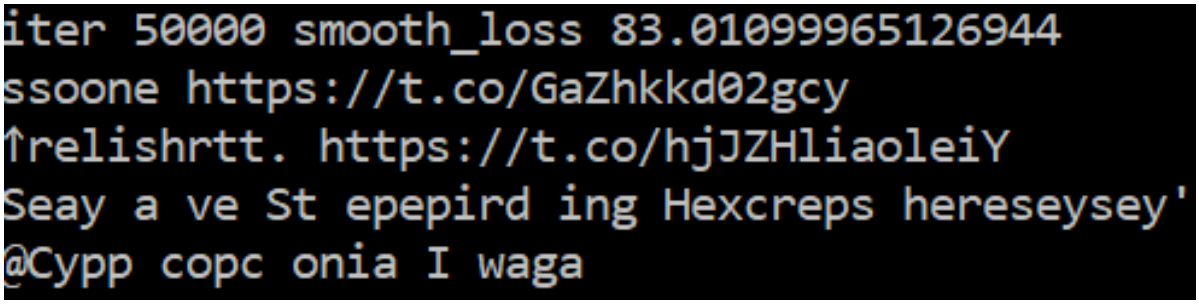
Figure 3: seq=40 Loss



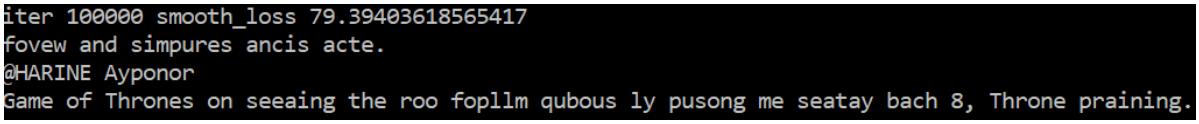
(a) iter0



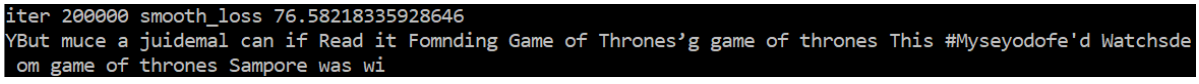
(b) iter10000



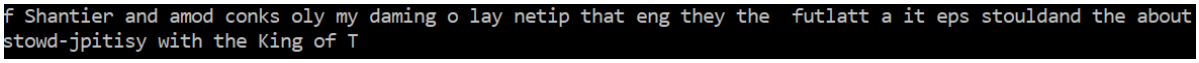
(c) iter50000



(d) iter100000



(e) iter200000



(f) final

Figure 4: Synthesized Text Seq=40