Merilin Sousa Silva, Shubhi Pareek
Machine Translation
Department of Computational Linguistics

FS 2025
UZH

MT Exercise 2

Topic: RNNs and Language Modelling

Due date: Tuesday, 1 of April 2025, 14:00

Starting remarks:

1. The forked github repository is as follows: https://github.com/merilinsilva/msousa_spareek_mt_exercise02.git

2. Everything was run from the mt-exercise-02 repository and all instructions are in the README.md file!

1. Training a recurrent neural network language model

a. Present your chosen data, does it have any special attributes that you expect to have an influence on the text generation of the model?

   i. We chose to combine multiple classics from Gutenberg (English or translated into English), since we found that using a smaller dataset often led us to almost uninterpretable samples and the training would take a maximum of 40 seconds. Thereby, we increased the data size and adjusted the ratio of training, test and validation set accordingly.

   ii. We combined the text of 8 books:
      1. Dracula
      2. Emma
      3. The Strange Case of Dr. Jekyll and Mr. Hyde
      4. Frankenstein
      5. Les Misérables
      6. The Count of Monte Cristo
      7. The Picture of Dorian Gray
      8. Pride and Prejudice

   iii. We saw that the raw text had 198464 lines, so we changed the division count between training data, test data and validation data. Depending on the line count it should now dynamically do a train = 80%, test=10%, valid=10% split.

   iv. We retained the set parameters such as vocabulary as instructed and thereby the vocabulary was set to 5000. When we looked at the preprocessed data, we noticed that a lot of the words were replaced by the special token <unk>. Our hypothesis is that this happened, since the raw text contains a large vocabulary with also different character names and thus a “small” vocabulary will lead to a lot of replaced words. This vocabulary count, like mentioned in the explanation on the assignment sheet, sets the limit to which words will be used for training, depending on their frequency rank. Since our training data is from Books, the word ‘chapter’ and numbering appears often and was thus considered as an element in the vocabulary. However, we think that this word does not hold much significance and will thereby not provide much information to the model during training. Therefore, we tried removing this word and trained the model again without it.

v. Changes:
   1. Before replacement:
      | End of training | test loss 4.29 | test ppl 72.88
   2. After replace:
      | End of training | test loss 4.27 | test ppl 71.65
   3. Conclusion: By removing the phrase “CHAPTER” + roman numeral/-s, the perplexity score decreased, indicating that the model got better at predicting future words.

b. Take a look at the sample generation, what are your impressions?
   i. Sample without removed “CHAPTER”:
      1. *works by his revenge . ” “ Nor ! ” “ I am a endeavour , ” said Mademoiselle Gillenormand, in conscience , and everywhere ; ran out up the eternal sign of a more elegant body . <eos>“ M. Hurst . <eos> <unk> hearts until ready thy <unk> ’ s <unk> , I continued , a nakedgardener to which the funeral is like <unk> and <unk> <unk> , a good <unk> <unk> that <unk> she is a great <unk> to stop at ignorance , hardly by it , then my nails at obedience . <eos> Dr.*
         a. Subsequent to the model learning words that are out of vocabulary boundaries as the token “<unk>” and that token appearing quite often, one can see that this token was generated multiple times in the sample. This was, thereby, expected, since that token probably has a high probability of appearing after different preceding words. To avoid such an overuse of this token, one would have to extend the vocabulary size (currently 5000).
         b. The second thing that we noticed was that in the generated text the special end-of-sentence boundary token ‘<eos>’ was generated. We assume that this token was inserted before training to allow the model to

learn where sentences end, which can improve its understanding of syntax and position during training and generation.

c. The use of punctuation seems rational, which is another positive aspect of this generated sample.

d. However, while some sentences are somewhat coherent and follow the jargon of 19th literature classics, most of them seem like gibberish. Perhaps, by lowering the temperature, it could lead to a more comprehensive sample.

e. Finally, there is a cut off point to the generated text, which can be seen in the last sentence, where it was cut off after two tokens.

ii. Sample with removed “CHAPTER”:

1. *. <eos> Life concluded a regard to be like him to contain about their catastrophe , which am silent .*
<eos> While much <unk> and prodigious forces seemed reverend of the smile in its death , to become properly outer, once still <unk> , since his wit will not recollect it , with ourselves , immediately not <unk> , but his father , I was began to time . <eos> And she said . <eos> It would not be at his own to keep him he is always found it , as she did at me . <eos> On

a. After removing the phrase, ‘<unk>’ appears way less, likely because it reduced noise and discontinuities in the data, allowing the model to focus more on the actual narrative and sentence flow, rather than being disrupted by section markers.

b. The output still contains some syntactic and semantic peculiarities, yet the structure seems more coherent and the use of POS makes more sense than before.

2 Parameter tuning: Experimenting with dropout

a. Can you see a connection between the training, validation and test perplexity? Based on your results, which dropout setting do you think is the best and why?

i. Something that is clearly visible from the start is that the model was overfit and contains a lot of oscillations. We assume that the cause of it lies in the little amount of data and the high starting learning rate used, even if it is reduced in a schedular mode, it may still be too high and then the learning rate gets to zero prematurely (this leads to no learning steps being made in the last 2-3 epochs).

ii. The validation, training and test perplexities all indicate the same thing – the order of perplexity reduction rate. The model with the dropout value 0.8 performed the worst. In the test perplexity it had the highest score (131.24) showing the highest surprisal, in the validation perplexities it started with the highest score and plateaued the quickest (also a sign of overfitting) and in the training perplexities, the values reduced quickly, but also plateaued the quickest. This was to be expected, since the dropout is too high, leading the model to learn almost no nodes. The model with the dropout 0.0 performed the best in our opinion, since it reaches the lowest perplexity value and plateaus the latest in the time frame of epochs. Are assumption is that the dataset is too small to add dropout to it. Perhaps adding it causes more harm, since there is not enough information in general for nodes to be deactivated (set to zero).

b. Sample some text from the model that obtains the lowest test perplexity, for instance by changing the script scripts/generate.sh. What do you think of its quality? Does it resemble the original training data? Sample some text with the highest test perplexity. Can you see a difference to the lowest scoring one?

   i. When we sampled a text with the model_0.8.pt (see 'samples/sample_0.8_dropout), we noticed that the vocabulary does resemble the training data in regards to the jargon being of the 19th/18th century classic literature genre, however, the text lacks heavily in fluency, semantics and syntactics.

   ii. The second sample (see 'samples/sample_0.0_dropout) has the same '<unk>' frequency, yet is more fluent and its semantics are more comprehensive, which supports our initial thought of it having the best model (despite it being overfit).