ECE 467: Natural Language Processing - Final Project

Bonny(Yue) Wang

Library Used

TensorFlow

Task

Text generation.
Generate Chinese poems from Tang Dynasty

Dataset

From https://github.com/chinese-poetry/chinese-poetry

In the JSON folder:

Poet.tang.0.json
Poet.tang.1000.json
Poet.tang.2000.json
Poet.tang.3000.json

Are arbitrarily selected to be the datasets of the project.

Split Datasets

Datasets are splitted into batches of size 64

In each batch, there is a pair of sequence of length 100 with one character shifted function as the input and the target(prediction).

Since the task is generating text. There is no need for a testing dataset. However, the start phrase is chosen to be ‘title’ based on the format of the dataset and anticipated format of the result.

Architecture

RNN with GRU is chosen for this project.

Compared to LSTM GRU has fewer parameters and thus is easier for training. Based on the size of data and the limit of the training environment, GRU is chosen for its computational efficiency.

GRU also prevent the vanishing gradient problem in traditional RNN.

The input layer of the project is the word embedding. The GRU layer is in between the input and the output layer. The output layer will produce the log likelihood of all the unique characters and the one with maximum possibility will be chosen to be appended to the result.

The size of the input layer is 1*5129(the size of the vocab), same as the output layer. The size of GRU layer should be 64 since the batch size is 64.

The word embedding dimension is chosen to be 1024 to create more sparse matrix since there are much more unique characters in Chinese(comparing to English).

Epoch 20 is chosen to produce good results. The loss value has a clear reduction after each epoch. However, it does have the cost of long running time.(Since I have a rtx2060 on my laptop, the training time is reasonable. It could take much longer time on other machines without GPU acceleration). In addition, by experimenting, the loss value is below 2 after 16 epochs, below 1 after 26 epochs, below 0.5 after 32 epoches and around 0.2 after 40 epochs.

Result

As a person who did not have too much in depth knowledge in Chinese poetry from the Tang dynasty, the result looks decent to me. Some of the sentences sound like a real poem. It also figured out the author correctly. Since the title and the name of the author have more clear patterns, the results for these two parts are generally reasonable. Most of the author names generated are actual authors instead of a random name. Most of the titles are also similar to an actual title or is an existing title.

However, there are more problems in the paragraphs part. Usually, if most other sentences have the same size, there should not be an exception. We can see several outputs have different sizes for sentences. The last poem of the sample output also have “paragraphs:張說” followed by “paragraphs:” which is clearly not right. It could be inferred that the correct new line character is not generated and the “張” is a very typical last name so instead of paragraph, the author name is generated.

Sample Outputs:

title:橫吹曲辭 巫山高
authors:杜甫
paragraphs:

幽州意氣箕山坂，戰子三蕃阻鼠鞭。
躍壁月中不改，金事不齊身。
草木相微滿畫堂，繡蒿金縷繡林塘。
琱青綺閣揖如已，一時榮落自君心。
專詔燕樓崇已識，城梁和洗絕氛租。

title:白雪寄王師秋夜懷徐公挽歌詞二首 二
authors:李賀
paragraphs:
九月笙歌鼓吹笙，蝶動風竿過上驄。
蹌纖塵霧兮兩殷獨感，不同歲李之明驕。
半月羅輪起曉天，明月明珠照耀雲。
歌宛轉歌崔玉幣，河山桂族垂雲湖。
小姬塞口入張舍，玉檻浮雲臥未央。

title:橫吹曲辭 驄馬千堆
authors:郭震
paragraphs:
隴火無氣色，山空髮盤乾。
深沈不可巡，知食不能聽。

title:相和歌辭 大歌辭 鳳吹曲
authors:沈佺期
paragraphs:張說
paragraphs:
鴛鴦夜白水驚碧，噴潤清笳簇金樞。
綺房鶴容如不息，幽源今萬旗列出。

**This project is highly based on https://www.tensorflow.org/tutorials/text/text_generation