Hi, I am Xinyi, Today I am going to show my midterm project, text generation using LSTM.

When I started to do the project, I found some of you already post the project in the spreedsheet, and there are a lot of topics about classification. Then I thought, whether I could do some other interesting stuff, and thought of the LSTM model, based on some given text, more related text could be generated.

Here is the original dataset, some text from a novel call wonderland. And I also deleted some chapters of the book and got a smaller dataset.

And my goal is to learn the sequence of the characters from the txt file, the novel and generate some text.

Step1: read the dataset, and we should notice one thing is that we do not care the uppercase and lowercase, they are the same vocabularies, so I convert them to the same lowercase. And I sorted the unique characters in the text, so that there were 47 characters finally. And each of them has a unique number.
Then we started to learn the dataset, how to learn? it might be difficult to learn the original code, so that I created some simply version to better understand. For example, I read from the second characters to the fiftieth. and the Y is the fifty first, because I know the unique number for them, so I changed the characters to numbers.
SO we can understand the difficult one, read 100 characters at the same time and the one hundred first is Y. Then move forward to one step per time, until all the text has been read.
rashape and normalized X. As for Y, I modified the code here, because the source code provided an inaccessible code, the np_utils package could not be used now.

Step2:Modelization
I did LSTM for another time, changed the drop out rate to make the model learn more complicated model, and changed the number of epoch and batch size because the dataset the model is too complicated I wanted it learn a bit faster.
Even though I changed, I still ran for more than three hours after using GPU. softmax has to be used because it can count the probability of each class, each potential output

In the source node, the text generation is to get a random seed number firstly, and use the seed number getting back to the dataset, and get the text for the dataset and generate character based on that.
I changed it to input some text ourselves, and generate character based on the input. I would like to get such expectation, like when you write some words in email, they could guess what you might type write based on the text you already wrote.
So that input text is not random. I did that, changed the simple dataset, and found after input, no more output except for the input words.
So, I went back the original dataset and found it could generate some thing based on the input , although it has a lot of duplicate. It might be caused by my model is not complicated enough.
And the source code still have trouble, which might be difficult to avoid.

I think it is a good try to get the logic of the text generation processing and learn how to use the LSTM.