Logo

Description automatically generated with medium confidence

|  |
| --- |
|  |

Last edit as of 11/23/2021

Version 1.0

Yupeng Cui

*As of 07/2014*

**Purpose:**Indicate the sentence which shows a cancer pathology in a medical note

**Result**: Since we want to get as higher as a recall rate with a normal precision, the current model is training on 49716 data points and testing on 12429 data points with 12354 label 0 sentences and 75 cancer pathology sentences. With the model, we can get 0.96 recall and 0.94 precision.

|  |  |  |  |
| --- | --- | --- | --- |
| precision | recall | f1-score | support |
| 0.999757 | 0.999595 | 0.999676 | 12354 |
| 0.935065 | 0.96 | 0.947368 | 75 |
| 0.999356 | 0.999356 | 0.999356 | 0.999356 |
| 0.967411 | 0.979798 | 0.973522 | 12429 |
| 0.999367 | 0.999356 | 0.999361 | 12429 |

**Please note: Since the number of label1 records in testing is only 75 which is a small number, the recall here 0.96 is not a highly reliable number since the variance is too large. But from the results we can see the model does have strong ability to predict label1 data.**

Data Engineering Process Overflow

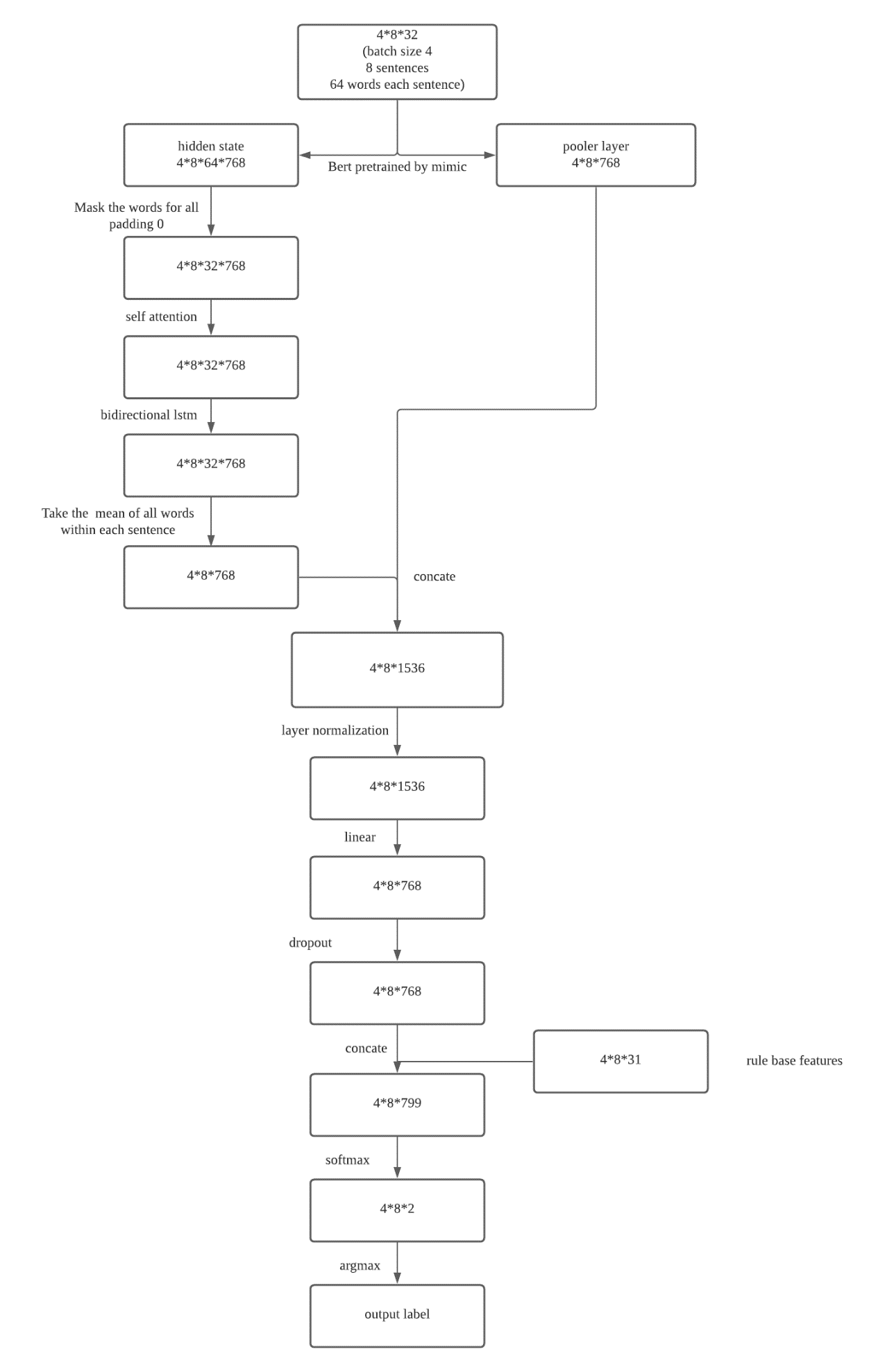
The input data is a json file which contains the id, patientid, note, label, and the encounter\_date.

1. Manually label the note data and add <start> and <end> to the both sides of the cancer pathology sentence.
2. Extract the id, patientid, note, label to the excel file (sample: cancer.csv)
3. Separate the label0 note into sentences and label as 0, keep label1 note as original, drop duplicated notes (cancer\_sep.csv)
4. Create training and testing data using dedup file (sample: train.csv,test.csv) with 0.2-0.8 ratio

**Please note: Step 1 is not in the code yet. The input file for training process is cancer.csv**

Model Structure

1. Put the sentences into bert model pretrained by mimic database emilyalsentzer/Bio\_ClinicalBERT
2. Mask the hidden state layer
3. Add a self attention layer
4. Take the average of every word in a sentence
5. Merge the pooler layer from bert output
6. Bidirectional LSTM
7. Layer Normalization
8. Dropout
9. Add rule based features (We have some keywords indicating whether the sentence is cancer pathology or not)
10. Softmax



Training Output

1. Run main.py –do\_train –do\_eval
2. The output will contain 3 files: model, performance metrics, prediction on all testing data

Model Inference

1. Put json files in SenTag\_serving\data\eval\_data
2. Run run\_inference.py
3. The model inference input is the original json file which encounter date and note
4. Output will contain the encounter date, sentences which are predicted as cancer pathology, whole note which contains the cancer pathology sentences