Data preprocessing and modelling is done on google colab.

**Model used:**

1. Logistic Regression
2. Naïve Bayes
3. SVM
4. Bert (Best model)

**Model Results:**

|  |  |  |
| --- | --- | --- |
| Model | Train Accuracy | Test Accuracy |
| Logistic Regression | 99% | 93% |
| Naïve Bayes | 79% | 83% |
| SVM | 97% | 87% |
| Bert | 99% | 99% |

Bert is giving good result in both train data and test data where Logistic regression and SVM perform well on train data but is performing slightly reduced on test data.

Lets focus on bert model.

# BERT

* Load data to colab
* Loading Chinese stopwords(downloaded from internet) and clean the data
* Select bert model, Here I have used – bert-base-chinese
* Load bert tokenizer
* Tokenize sentences

Hyper parameters:

* Max sentence length = 396 (median word length)

(We can also try max length of 128, 256, 512)

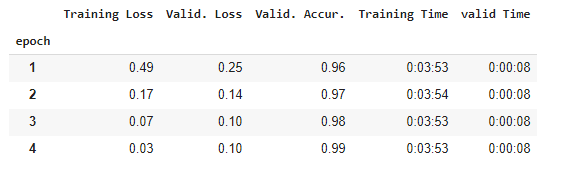
* + Padding: “post” (will add 0 at the end of sentence with length less than 396)
  + Truncate: “post” (will remove last part of the sentence after 396th word
* Prepare attention mask vector (To differentiate actual word and padding)
* Divide data in train and validation (90% train and 10% validation)
* Train BertForSequenceClassification

Hyper parameters:

* + Batch size: 4 (In bert paper, suggested batch size was 16 or 32) (But due to colab limitations I was not able to use large batch size)
  + Optimizer: AdamW
  + LR: 5e-5 (In bert paper, suggested LR was 5e-5, 3e-5, 2e-5)
  + Number of epochs: 4 (In bert paper, suggested num of epochs was 2/3/4)

We can do hyper-parameter tuning with these parameter (if required)

* Save model weight files and vocab json file to use it for inference
* Model results



* Inference code for model to predict on one news
  + Load the model weights and vocab json
  + Preprocess sentence (Remove punctuation and \n character like we did before training)
  + Tokenize sentence
  + Predict class

Results of inference code

I have taken few random news in Chinese and tested model on news. It gave accurate results.

