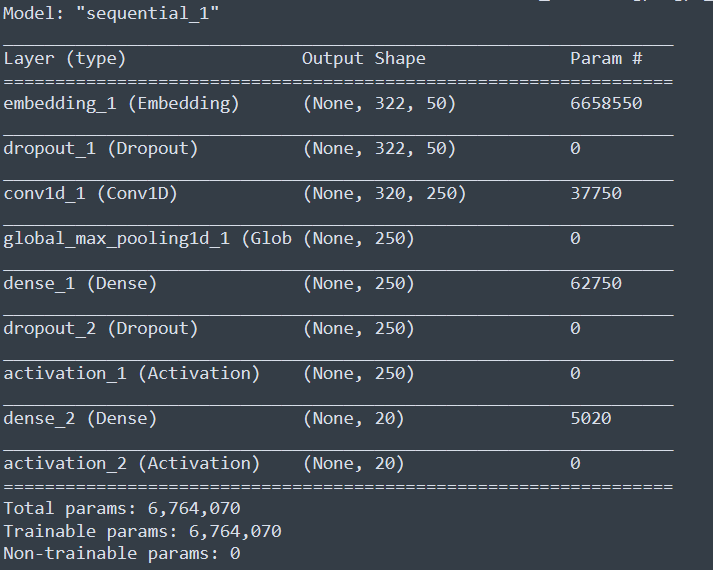
**Documentation**

The problem is multi-class text classification. There are 20 classes in total. The documents under each subfolder belong to the same class. The first step is data preprocessing. The purpose of this step is to generate training and testing dataset in the proper format that can be directly feed into a machine learning model. I choose to use Tensorflow to build the text classifier. Thus, it requires the input data to be Numpy arrays. The code “preprocessing.py” reads the documents data and outputs four Numpy arrays namely x\_train, y\_train, x\_test, y\_test.

In the preprocessing procedure, each document is first transformed into a sequence of words. This step is implemented by a word tokenizer. Then, each word is mapped to an integer based on a vocabulary dictionary. Finally, each document is represented as a sequence of integers. Each class label is transformed into an integer following the same manner. Next, the document and label lists are transformed into Numpy arrays.

The second step is to build a text classifier. I choose to use 1D-convolution network as the classifier. This type of network keeps the sequential information in the text documents and is widely used to solve the text classification problem. The figure below shows its structure.

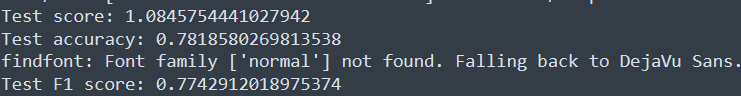


The first layer is the input layer. Each document is represented as a sequence of integers. The second layer is an embedding layer. Each word is transformed into a vector of 50 elements. The third layer is a dropout layer to prevent overfitting. The next layer is an 1D convolution later. After that there is a global max-pooling layer for dimension reduction. On top of it there are two dense layers. The second dense layer finally outputs the predicted class label, which is represented as a vector of 20 elements.

The model is trained for 10 epochs until its performance converges. The figure below shows the training status:



After training, the model is evaluated on the testing data. And the evaluation metrics are calculated.



The macro f1 score on the testing set is 0.77.