Report for NLP First Practice

Shanglun Wu
rmhiswu@ucl.ac.uk
Institute for Health Informatics
London, United Kingdom

1 INTRODUCTION

The key point of the dissertation is Natural Language Processing (NLP) and the first exercise before the main work is to be familiar with the NLP pipeline, which is a set of steps followed to build an end-to-end NLP project.

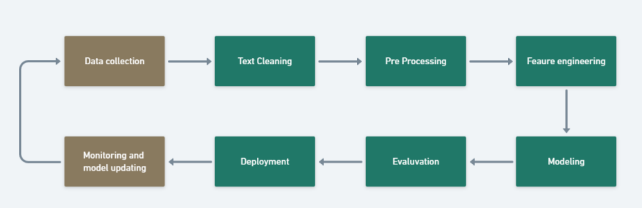

Figure 1: An end-to-end guide on NLP Pipeline

In this report, a text-based dataset was given and the process of Text Cleaning, Pre Processing, Feature engineering(word embedding), Modelling, and Evaluation will be covered.

2 DATA EXPLORATION

There are 27073 useful sample data in the data with missing text was deleted) and classified into 21 different fields.

label	
Conjunctival Diseases	750
Corneal Diseases	1756
Eye Abnormalities	859
Eye Diseases, Hereditary	1245
Eye Hemorrhage	126
Eye Infections	1149
Eye Injuries	435
Eye Neoplasms	917
Eyelid Diseases	1998
Lacrimal Apparatus Diseases	596
Lens Diseases	828
Ocular Hypertension	577
Ocular Motility Disorders	2508
Optic Nerve Diseases	1809
Orbital Diseases	2776
Pupil Disorders	188
Refractive Errors	466
Retinal Diseases	4056
Scleral Diseases	201
Uveal Diseases	1291
Vision Disorders	2542

Figure 2: Sample data distribution

As the above picture shows, there exists an imbalance between the classes. However, the size of the whole dataset is not large, oversampling or under-sampling could cause reducing the dataset size, and too small a dataset could not be used to train an accurate model.

Thus, although the data is imbalanced, changes should not be done currently.

3 TEXT CLEANING

For the text cleaning part, the prime target is removing the meaningless symbols. For some emotion-analysis datasets, some symbols with mood trends like: '?', and '!' are meaningful and should be kept for the following evaluation. While for the given dataset, the symbols have no effect on predicting. Thus, the punctuation in the string library can be used to filter the symbols.
The output of preprocess_symbol function returns the string from the document without symbols.
To be mentioned, this step only can filter the English symbol, the other language symbols (like Chinese symbols) might be considered in future work.

4 PRE-PROCESSING: TOKENIZER & STOPWORDS

The tokenizer takes the responsibility of transforming the text into a list of meaningful words. During this process, dropping meaningless words like: "is", "a", and "for" can effectively reduce the embedding(vector transforming) and evaluation time and increase accuracy by reducing bias from noise.

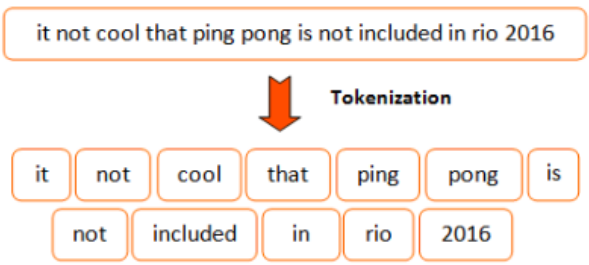

Figure 3: Tokeniser in NLP

In the tokeniser function, the text output by the preprocess_symbol function was first transformed to lowercase and split into a list of words. Then use the general English stopwords dictionary from nltk

to drop the meaningless words that matched in the dictionary. Finally, the function output the list of word for following embedding process.

5 WORD EMBEDDING

After the pre-processing, each sentence in the document was divided into arrays of meaningful words, word embedding here is to transfer words into vectors, which can be treated as input to train a supervised model. The word-to-vector model used here is GloVe, which contains 40000 dictionary words and for each word, it has a corresponding 50-dimension vector. For the words in the document which are not in the model dictionary, random vectors between -1 to 1 were assigned to them. However, the selection of the method for forming sentence vectors from word vectors needs to think carefully.

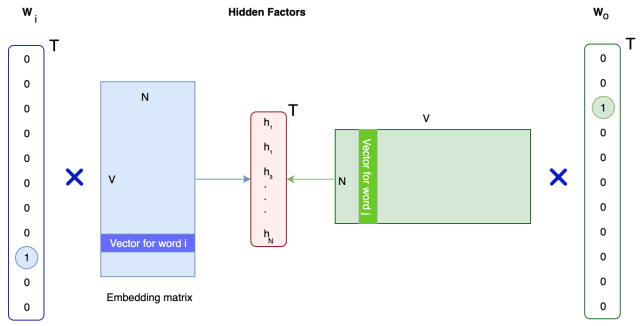

Figure 4: Word embedding by GloVe

At first, I calculated the average values of word vectors and treated them as sentence vectors. This method could ignore the length of different sentences. However, input vectors produced by this method led to quite a big difference between training and test accuracy. which might cause over-fitting. And for LSTM, the equal-length sentence might have a better result.

Thus, to solve the problem of different sentence sizes, the max sentence size was set as a parameter, and for each sentence, the vector is a two-dimensional array, which shape is max-sentence-length multiplied 50 dimension vector (GloVe for each word). To be mentioned, for the sentence that length is less than pre-set length, the loss word vector would be filled with zeros.

6 MODELLING

6.1 LSTM

For the model development, the following steps are taken:

SpatialDropout1D(rate=0.2): This version performs the same function as Dropout, however, it drops entire 1D feature maps instead of individual elements. 0.2 is a proper drop level.

LSTM with 64 hidden layer sizes, general dropout 0.2, and recurrent dropout 0.2, which is the proportion of neuron disconnection that controls the linear transformation of the cyclic state.

Dense(Fully Connection layer in Keras), with 21 units output, which corresponds to 21 label classes. And chose softmax as activation. Softmax can normalize a value vector to a probability distribution vector where the sum of probabilities is 1. Softmax here is used as the final layer of the neural network for the output of multiple classifications. The Softmax layer is often used in conjunction with the cross-entropy loss function.

Adam was chosen as an optimizer, which takes the advantage of adaptive learning rate gradient descent algorithm and momentum gradient descent algorithm and can not only adapt to sparse gradient, but also alleviate the problem of gradient oscillation. In the basic classification task, adam has a fast and accurate performance.

7 EVALUATION

For this exercise, the main purpose is to realize the whole NLP process, not to produce the most perfect model or most accurate hyper-parameters. Thus, with 20 epochs, and 128 batch size, the following figure shows the increasing step of training accuracy by epoch.

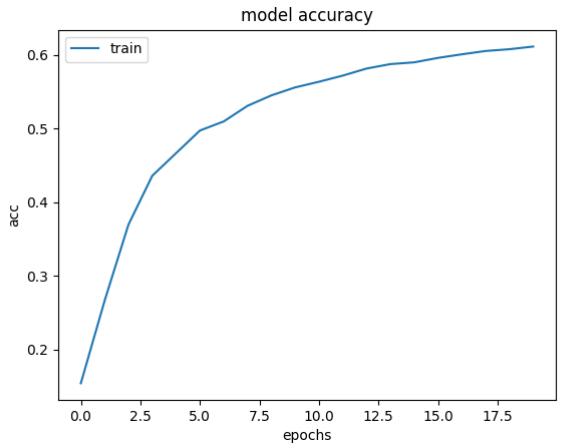

Figure 5: Train accuracy by epoch

With the current model, the final accuracy of training is around 67% and test accuracy is around 65%.

8 DISCUSSION AND REFLECTION

For this practice, basic hyper-parameter adjusting was applied. If necessary, k-cross fold validation would be used to adjust the hyper-parameter better, Beside the training and test dataset, split another validation dataset to improve the model performance.

And if the size of the dataset could be larger, over-sampling or under-sampling methods could be used to solve the imbalanced classes problem, which can increase the accuracy of the whole model.

For the word embedding process, accumulating the words vectors with the given length for the sentence vectors might increase the training speed (fewer dimensions), but might cause overfitting. If given more time, this method would be tested.

For the next step, I am using leisure time(new term beginning) to find whether a simpler model - CNN, could have a satisfying performance or not. If any results are developed, updates will be loaded on this repository.