Design and Implementation of Application Classification Based on Deep Learning

WenChuan Yang, Qiuhan Zhao, Rui Hua

Beijing University of Posts and Telecommunications

Abstract

This paper uses a deep learning-based model to solve the problem of automatic classification of mobile applications. In this paper, we address the classification problem of mobile applications from the perspective of text classification. By analyzing the major mobile phone application markets, we have developed the main categories of applications, and crawled the descriptions of various mobile phone applications as needed. With analyzing the original corpus of the crawl, the semantic information is further expanded by using data augmentation methods based on both word and char. Then, we design different text classification networks and compare the experimental results, and finally select the network with the best classification effect for tuning. The results of experiments show that the classification network of Bert+Highway+GRU designed in this paper has better classification effect. The average P/R/F1 value of the classification is 0.8820/0.8892/0.8856. The classification indicators under the above all reached 0.85 or higher, which in the first level label of those applications; at the same time, it also showed better performance in network training and convergence speed. The deep learning-based mobile phone application classification network designed in this paper has high classification efficiency and can achieve higher classification accuracy.

Methodology

1 Mobile Application Data Collection

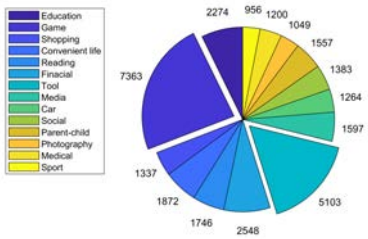

Fig. 1. Data Overview

This paper limits the data set to the common Chinese application market, such as Tencent, 360 and so on. With synthesizing the mainstream application market classification methods, we limit data sets to 14 categories: 'Education', 'Game, 'Shopping', 'Convenient life', 'Reading', 'Financial', 'Tool', 'Media', 'Car', 'Social', 'Parent-child', 'Photography', 'Medical', 'Sport', which are called first-level labels. Each first-level label contains a number of small classes, which are secondary labels. A total of 31249 crawling applications were crawled, and the specific distribution as shown in Fig. 1.

In addition, in view of the imbalance of data, we use the means of several data augmentation to adjust, data augmentation methods vary according to word and char forms. Word-based methods include deletion, random scrambling and random substitution.

TABLE I. THE RESULT OF DATA AUGMENTATION

Standard	Action	Result
Word	delete	a cat lies on the floor
	Substitute	a cat lies 0 the floor.
	Swap	on the cat the floor a lies
Character	insert	a cute cat lies on the floor
	substitute	a dog lies on the floor

2 Feature Model

Based on the previous experience, we choose four networks with better text classification results: TextCNN as a baseline, RNN structure based on hierarchical Attention, RCNN network, which is better than a single model in text extraction, and BERT model proposed by Google.

We use the pre-training version of 'Bert-base Chinese' provided by Google to train mobile application corpus. At the same time, in order to prevent the network from being too deep and alleviate the gradient problem, we superimpose two Highway layers on Bert's output layer. Then, context semantics are further correlated through a bi-GRU layer, and feature extraction of mobile phone applications is completed through the stitching and integration of the full connection layer. Finally, the classification prediction is completed by Sigmod or Softmax.

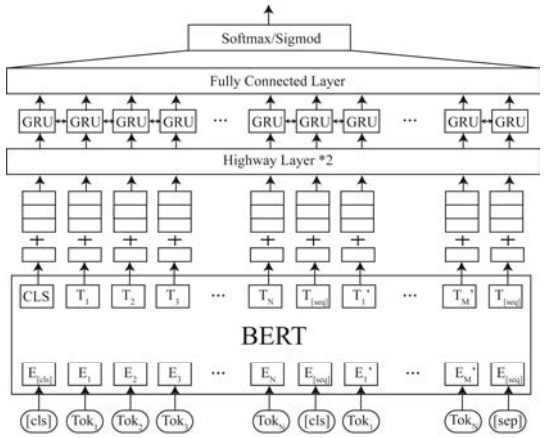

Fig. 2. BERT+Highway+GRU

Experiments

For each of the models mentioned above, we use a two-layer fully connected layer plus a Softmax function for classification. Moreover, we add a layer of Dropout between the two fully connected layers and set the loss rate to 0.5. The results of different deep learning models are shown as follows:

TABLE II. BEST ACCURACY OF DIFFERENT MODELS

Model	Accuracy	Precision	Recall	F1
TextCNN	0.8180	0.7922	0.7943	0.7932
RCNN	0.8700	0.8165	0.8024	0.8094
RNN+attention	0.8750	0.8555	0.8538	0.8546
BERT	0.8915	0.8741	0.8884	0.8812

Moreover, we make further simple parameter optimization for BERT. We evaluate the hidden layer parameters of the bi-GRU used by the BERT and the fully connected layer pool method. Among them, the original model uses 128 hidden states. By controlling a single variable, the number of different hidden layer parameters and different pool methods are used to further improve the classification accuracy.

TABLE III. Different Hidden States and Pooling Method

	Accuracy	Precision	Recall	F1
Hidden States				
128	0.8915	0.8741	0.8884	0.8812
200	0.8933	0.8827	0.8875	0.8851
256	0.8921	0.8800	0.8831	0.8815
300	0.8902	0.8716	0.8805	0.8760
Pooling Method				
Max-pool	0.8915	0.8741	0.8884	0.8812
2-Max-pool	0.8928	0.8744	0.8887	0.8815
3-Max-pool	0.8911	0.8728	0.8830	0.8779
Average-pool	0.8887	0.8722	0.8798	0.8760
Best Result				
	0.8936	**0.8820**	**0.8892**	**0.8856**

Conclusion

This paper analyzes the mainstream mobile phone application market, sets the classification criteria, and crawls the mobile application data according to the standard. After preprocessing and data augmentation of the original corpus, we input the data into different neural networks for comparison. Finally, the model of BERT+Highway+GRU got a higher score. We applied it to the classification of the original corpus after tuning, and achieved the average score of 0.89, which shows a good classification effect.

In addition, this paper analyzes the shortcomings in the experimental results. How to deal with the data imbalance in the original corpus, and how to further distinguish the cross-term, those will be the next research work to be carried out in this paper.