Automatic Tagging of Zhihu Questions

A Preprint

Yong Hu
Department of Computer Science
Beijing Institute of Technology
Beijing, China

Tan Yan
Department of Computer Science
Beijing Institute of Technology
Beijing, China

Chenyang Lyu
Department of Computer Science
Beijing Institute of Technology
Beijing, China

February 26, 2019

Abstract

In this study, we first give an overview for the shared task at the CCF Conference on Natural Language Processing & Chinese Computing (NLPCC 2018): Automatic Tagging of Zhihu Questions. The dataset is collected from the Chinese question-answering web site Zhihu, which consists 25551 tags and 721608 training samples in this shared task.This is a multi-label text classification task, and each question can have as much as five relevant tags. In this study, we propose a novel model to make auto-tagging to Zhihu questions and have a better performance over baselines. Our best performance among three proposed models is precision of 32.5, recall of 46.2 and F1 of 38.2, and those results are produced by RCNN model.

1 Introduction

The task aims to tag questions in Zhihu with relevant tags from a collection of predefined ones. This is a multi-label classification problem, several tags can be relevant to a given question. With the rise of social media, the text data on the web is growing exponentially. Furthermore, the label space is relatively huge compared to traditional text classification tasks. Make it is impractical for a human being to accurately assign tags to all those data. Machine learning methods are quite suitable for this task, and accurate tags can benefit several downstream applications such as recommendation and search.

Formally, the task is defined as follows: given a question with its title $x_t = (x_{t_1}, x_{t_2}, ..., x_{t_n})$ and description $x_d = (x_{d_1}, x_{d_2}, ..., x_{d_m})$, where x_{t_j} denotes the jth word in the title. The objective is to find its possible relevant tags in the predefined tag set. More specifically, given a specific tag tag_i, we need to find a function to predict whether tag_i is relevant to the current question with title x_t and description x_d.

$$p(tag_i|x_t, x_d) = f(x_t, x_d, tag_i, \theta) \tag{1}$$

where θ is the parameter of the function.

In this study, we propose a novel model which uses CNN and RNN to learn very rich representations of question's title and it's description, then we combine two representations and feed it into feed-forward neural nets and get final tags of this question. According to the experimental results, our RCNN model gets the best performance over BiGRU and CNN model, RCNN model performs near 2 points better than BiGRU model, 1 points better than CNN model for precision, recall and F1.

2 Data

In this task, data are provided into three part: training, development, and test. Each question in the dataset contains a title, an unique id and an additional description. The labels are tagged collaboratively by users from the community question answering web site Zhihu. To improve the quality of the data, we removed infrequency tags, and relabeled manually to build development and test dataset. There are 25551 tags and 721608 training samples in training data, 8947 samples in development data and 20597 samples in test data. Some samples from training dataset are shown in Table 1.

The dataset is different from widely used text classification datasets. Firstly, the label space is relatively huge and there is a data imbalance problem. Table 2 shows the statistics of the numbers of training samples for each label, we can see that almost 30% labels only have 5 to 10 training samples, while there still are some labels may have more than 5000 training samples. Secondly, the task is a multi-label problem and the number of labeled tags is not fixed for each question with a range from 1 to 5, Table 3 shows the statistics of the numbers of labeled tags for each question. Thirdly, since the dataset is collected from Zhihu whose contents are all generated by users, the text styles vary from user to user.

Table 1: Training samples from the dataset.

question_title	question_description	tags
How to keep others from affecting your mood?		mood, mood control
How does one create business plan?	I have some ideas,how can I get investment?	business plan, VC

3 Evaluation

For each question in the test set, the model is required to predict as much as five relevant tags, and the tags are sorted by their predicted probabilities. Specifically, the number of predicted tags for a given question can be less than 5 or even be 0 if the model can’t find enough relevant tags to the question.

The results are evaluated on the F_1 measure.We compute the positional weighted precision.Let $correct\_num_{p_i}$ denotes the correct count of predicted tags at position i, and $predict\_num_{p_i}$ denotes the count of predicted tags at position i.The precision, recall and F_1 measure are computed as following formulas:

$$F_1 = 2 \times \frac{P \times R}{P + R} \tag{2}$$

$$P = \frac{\sum_{i=1}^{5} correct\_num_{p_i}/log(i+2)}{\sum_{i=1}^{5} predict\_num_{p_i}/log(i+2)} \tag{3}$$

$$P = \frac{\sum_{i=1}^{5} correct\_num_{p_i}}{ground\_truth\_num} \tag{4}$$

Table 2: Statistics of the numbers of training samples for each label.

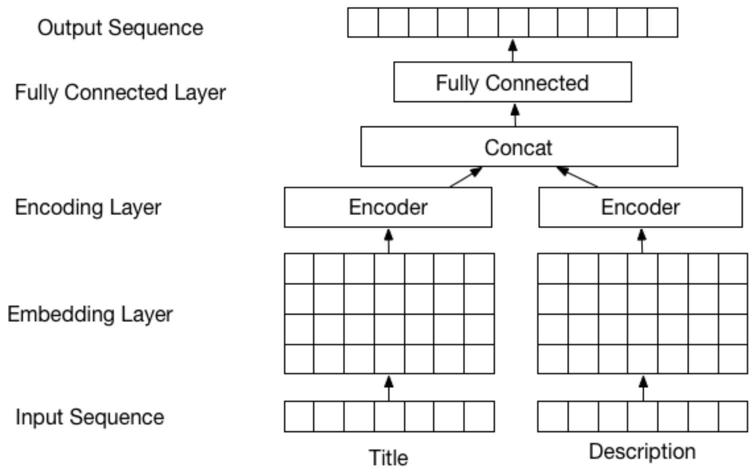
Number of training samples	5 to 10	10 to 50	50 to 500	500 to 1000	1000 to 5000	5000+
Count of labels	7651	11988	5158	422	296	36
Percentage of labels(%)	29.94	46.92	20.19	1.65	1.16	0.14

Table 3: Statistics of the numbers of labeled tags for each sample in training data.

Number of labeled tags	1	2	3	4	5
Count of samples	134190	123397	151553	143388	169080
Percentage of labels(%)	18.60	17.10	21.00	19.87	23.43

4 Model

In this study, we propose a novel end-to-end encoding-decoding model, which is shown as below:



The input of our model consists of two parts. The first part, which is the title of question, is defined as $(t_1, t_2, .., t_m)$.The second part, which is the description of question, is defined as $(d_1, d_2, .., d_l)$, where $t_i \in \mathbb{R}^{|V|}$,$|V|$is the dimension of word embedding. Firstly, we use embedding layer to encode title and description respectively, and then feed them into encoder layer to yield representations of both title and description. Then we concatenate the two representation vectors and pass the concatenation to decoder, which is a fully-connected neural nets. Finally, the output of decoder layer will be fed into a *sigmoid* function, and we can get $predict_y, predict_y \in \mathbb{R}^{|L|}$, where $L$is the length of tag.. $predict_y_i$is the probability that this question belongs to tag i. We use the output tag of our model and ground-truth tag to compute cross-entropy as out loss function.

For encoder, we conduct three state-of-the-art document classification model to perform feature extraction, including bidirectional-GRU+attention model, CNN model and CNN+GRU model.

4.1 BiGRU+Attention

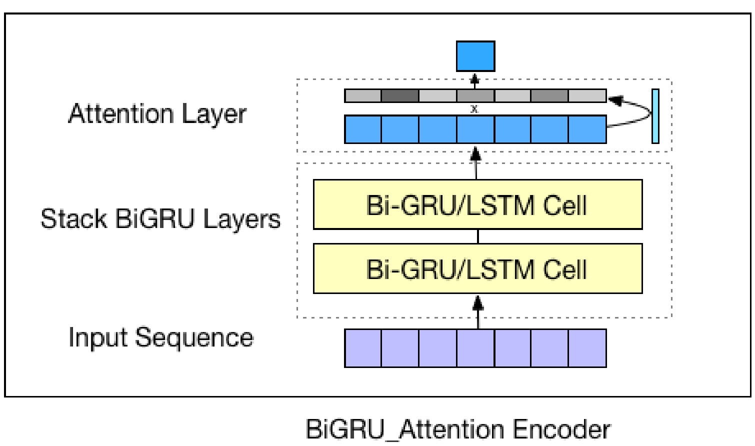


Figure 1: Bidirectional-GRU Attention Encoder

As shown in Figure 1, we first encode the sequence of word embedding $(x_1, x_2, ..x_n)$ by stacked bidirectional-GRU, yielding $(h_1, h_2, ..h_n)$. Then we calculate the importance of each h_i as following:

$$p_i = sotfmax(h_i W \dot{k})$$

Table 4: The hyper parameters of model

Max-Len Title	Max-Len Description	Word Dict Size	Word Dimension	Hidden-State Size	Layers of Decoder
30	200	19606	150	200	2

Where $h_i \in \mathbb{R}^{2*H}$, H is the size of hidden layer of bidirectional-GRU. We combined forward-GRU and backward-GRU, so the size of each hidden state is $2 \times H$, $W \in \mathbb{R}^{2*H,2*H}$ is a matrix, $k \in \mathbb{R}^{2*H}$ is a vector. Both are trainable parameters.

$p_i$is the degree of importance of token x_i, which is a scalar between 0 and 1.

The overall length of the whole sentence is $o = \sum_{i=1}^{n} p_i h_i$.

4.2 CNN

CNN(Convolutional Neural Networks) is a classical model for document recognition, which is proposed by Yann LeCun and is employed to perform hand-written recognition and image classification.This CNN-Encoder consists of two convolutional layers and a max-pooling layer. We first feed our input sentence into convolutional layer to extract features of input sentence, the advantage of stacked convolutional layer is that we can learn more abstract relationship of features.After convolutional layer, we pass the output of convolutional layer to max-pooling layer and finally get the top k tag of this question. In this model, we encode title and description with two different convolutional layers respectively.

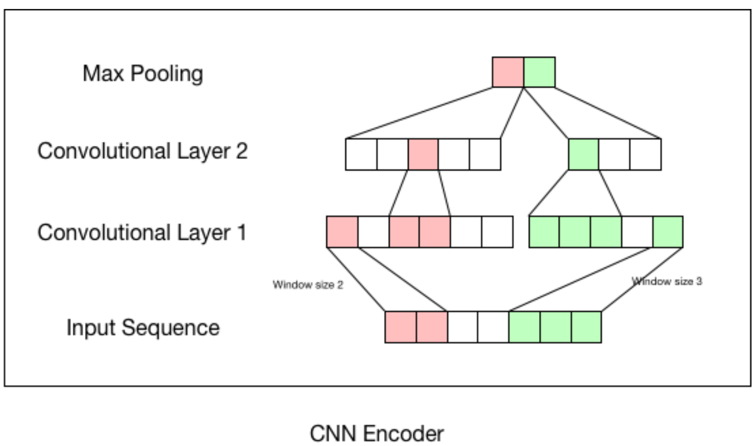


Figure 2: CNN Encoder

4.3 CNN+BiGRU

To address the problem that CNN and Bi-GRU cannot fully capture the most important feature and model the dependencies among input tokens, we combine CNN and Bi-GRU. In this model, we first encode input sentence using Bi-GRU or Bi-LSTM, then we concatenate the hidden state of each token. We add a residual connection that we concatenate raw word embedding and hidden state of Bi-GRU. After encoding phase, the representations of title and description would be fed into CNN, which will classify this question according to the representations. Finally, the output of CNN is the tags of this question. The sketch of RCNN encoder is shown in Figure 3.

5 Experiment

We conduct experiments of three model mentioned above, the hyper-parameters is set as shown in Table 4. Our project code is available on `https://github.com/nghuyong/ZhihuLabelsPrediction`

The result of experiment is shown in Table 5.

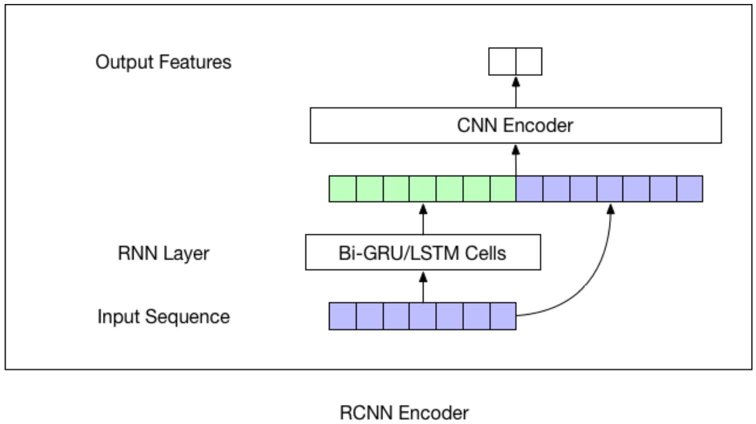


Figure 3: RCNN Encoder

Table 5: Results

model	precision	Recall	F1
BiGRU+Attention	30.8	43.5	36.1
CNN	31.8	45.3	37.4
RCNN	32.5	46.2	38.2

6 Conclusion

We can see from the evaluation results that the RCNN model gets the best performance over BiGRU and CNN model.The reasons behind that can be summarized to two points.First, RNN can learn a rich representations of title and description, attention mechanism can enrich the representations.Second, CNN is a strong model for feature extraction so that CNN can fetch the most relevant words for tag classification. RNN model has shortcomings that it cannot perform well on feature extraction, and learning representations is a weakness of CNN model. So either a single RNN-GRU/LSTM + attention model or a single CNN model cannot get a good result.Taking advantage of two models is a reasonable way.

We prepare to deploy transformer in the future work, which is proposed by Google and has the state-of-the-art performance on various NLP tasks, to our multi-tagging task.

7 Reference

1. LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.

2. Han, Eui-Hong Sam, and George Karypis. "Centroid-based document classification: Analysis and experimental results." European conference on principles of data mining and knowledge discovery. Springer, Berlin, Heidelberg, 2000.

3. Yang, Zichao, et al. "Hierarchical attention networks for document classification." Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2016.

4. Zhang, M.L., Zhou, Z.H.: ML-kNN: a lazy learning approach to multi-label learning, Pattern Recognition, 40(7), 2007

5. Tsoumakas, G., Vlahavas, I.: Random k-labelsets: an ensemble method for multi-label classification. ECML 2007

6. LeCun, Yann, and Yoshua Bengio. "Convolutional networks for images, speech, and time series." The handbook of brain theory and neural networks 3361.10 (1995): 1995

7. Zhang, Xiang, Junbo Zhao, and Yann LeCun. "Character-level convolutional networks for text classification." Advances in neural information processing systems. 2015.

8. McCallum, Andrew. "Multi-label text classification with a mixture model trained by EM." AAAI workshop on Text Learning. 1999.

9. Baker, L. Douglas, and Andrew Kachites McCallum. "Distributional clustering of words for text classification." Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998.

10. Luo, Xiao, and A. Nur Zincir-Heywood. "Evaluation of two systems on multi-class multi-label document classification." International Symposium on Methodologies for Intelligent Systems. Springer, Berlin, Heidelberg, 2005.

11. Yan, Yan, et al. LSTM: Multi-Label Ranking for Document Classification. Neural Processing Letters 47.1 (2018): 117-138.

12. Zhang, Yue, Liying Zhang, and Yao Liu. "Linked Document Classification by Network Representation Learning." Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data. Springer, Cham, 2018. 302-313.

13. Pivovarova, Lidia, and Roman Yangarber. "Comparison of Representations of Named Entities for Multi-label Document Classification with Convolutional Neural Networks." Proceedings of The Third Workshop on Representation Learning for NLP. Association for Computational Linguistics, 2018.