Enhancing Deep Learning-Based Multi-label Text Classification with Capsule Network

Siyi Yan
Internet of Things Engineering, International School, Beijing University of Posts and Telecommunications, Beijing, China
Email: ysy409@bupt.edu.cn

Abstract. Given a piece of text, multi-label text classification (MLTC) is designed to mark the most relevant one label or multiple labels for the text. Most of the existing MLTC models use convolutional neural network (CNN) as feature extractor, but CNN will lose information when dealing with MLTC task. In this paper, we explore the CNN combined with capsule network for MLTC. We use capsule network instead of pool layer in CNN to extract information related to classification results in high-dimensional features. We also explore the way of combining recurrent neural network (RNN) and CNN to model the characteristics of time and space for capsule network to complete classification. In two open MLTC datasets, our model achieves the better results as the baseline system, which shows the effectiveness of the combination of capsule network and CNN for MLTC.

Keywords. MLTC; CNN; capsule network; RNN.

1. Introduction
This Text classification is a basic task of text mining, including single label classification and multi-label classification. The former only assigns a label to a given document, while the latter classifies a document into different topics. In this paper, we focus on multi-label text classification, which has become one of the core tasks in natural language processing, and widely used in topic recognition, question answering system, emotional analysis and other tasks.

With the rapid development of big data, MLTC is facing a huge challenge because it has to deal with massive documents, words and tags at the same time. Therefore, it is urgent to develop an effective multi-label text classifier for various practical applications. Multi-label text classification allows multiple labels to exist in a single document, so there is semantic correlation between labels because they may share the same feature. At the same time, documents can be long, and complex semantic information can be hidden in noisy or redundant content. In addition, most documents belong to a few document tags, while a large number of “tail tags” only contain a very small number of positive documents.

2. Related Work
The Traditional MLTC methods include problem transformation and algorithm adaptation. The problem transformation method transforms multi-label classification into single label classification. The binary relation (BR) algorithm classifies each label as a separate category [1]; the label powerset (LP) method creates a binary classifier for each label combination [2]; the classifier chains (CC) algorithm classifies by simulating the correlation between labels [3]. Algorithm adaptation method extends the existing algorithm to make it suitable for multi-label classification tasks without problem.
transformation. Clare et al. [4] use ML-DT method to build decision-making based on multi-label entropy for classification; Elissée et al. [5] propose Rank-SVM and use support vector machine for classification; Lu et al. [6] use ML-KNN method to deal with big data set problem.

Both of the above methods are based on traditional machine learning methods and rely on a large number of feature engineering. With the development of deep learning, neural network methods are widely used in multi-label text classification tasks. Compared with traditional machine learning methods, deep learning methods can automatically learn text features and have more generalization advantages. At present, the mainstream neural networks include convolutional neural networks (CNN), recurrent neural networks (RNN) etc., which show good performance in different fields [7].

Deep learning is widely used in the field of natural language processing [8]. Compared with traditional machine learning methods, it does not need artificial design features, can automatically learn and extract text features, and has achieved better results than traditional methods. Zhou et al. [9] propose BP-MLL method, for the first time, use the neural network model for multi-label text classification, and use full connection network and sorting loss for classification; Jinseok et al. [10] use adagrad and dropout to accelerate convergence and prevent over fitting in the training process, using cross entropy function as the objective function; Berger et al. [11] capture the word order by the pre-trained word2vec word vector, which is used in CNN and GRU to directly use in multi-label text classification. Compared with the traditional word bag model, the performance of classification is improved; Kurata et al. [12] use the co-occurrence relationship between category labels to initialize the output layer of CNN to obtain the correlation between labels; Chen et al. [13] propose a new method for CNN and RNN In the fusion mechanism, which takes the output from CNN as the input of RNN to capture the semantic information of text, and then predicts the category; Nam et al. [14] first use the encoder decoder model in machine translation for multi-label classification, propose a formula for calculating the joint probability of labels, and generate classification labels in order; Yang et al. [15] first use reinforcement learning for MLTC, propose an encoder decoder model based on deep reinforcement learning, which regards class labels as sets to reduce the dependency between labels.

Most of the above models are based on CNN and RNN network structure. The pooling operation of CNN results in information loss, and RNN will generate gradient explosion or gradient disappearance problem with the increase of sequence input length. In order to solve the information loss caused by CNN's pooling operation, and make better use of the high-dimensional features extracted by deep neural network for output, we design a new network structure, using capsule network instead of the pooling operation in convolution neural network, in order to achieve better classification accuracy.

3. Model

3.1. Capsule Network

Traditional CNN uses pooling operation to reduce the computational complexity of convolution operation and capture local features, but pooling operation will cause information loss. In capsule network, neuron vector (capsule) is used to replace the single neuron node of traditional neural network. Each value is represented by vector. Dynamic routing is used to train the neural network to learn the relationship between words and cluster the feature vectors so as to obtain not only the position information of words in the text, but also the local space characteristics of the text. The dynamic routing process is shown in figure 1.
where $U_i$ is the input of the network, $V_i$ is the output of the network, $W_{ij}$ is the weight matrix, $C_{ij}$ is the coupling coefficient. Squash ($\cdot$) is the compression activation function, which compresses the vector to obtain the module length $V_i$ of the vector. Using the size of vector module to measure the probability of an entity, the larger the module value is, the bigger the probability is. The whole dynamic routing process is iterated $r$ times to obtain the local eigenvector $V_i = (V_{i1}, V_{i2}, \ldots, V_{in})$.

The calculation method is shown in equations (1)-(6):

$$S_j = \sum_j C_{ij} U_{ji} \quad (1)$$

$$U_{ji} = W_{ij} U_i \quad (2)$$

$$C_{ij} = \text{softmax}(b_{ij}) \quad (3)$$

$$b_{ij} \leftarrow b_{ij} + U_{ji} V_i \quad (4)$$

where the initial value of $b_{ij}$ is set to 0, and $b_{ij}$ is updated by dynamic route and $C_{ij}$ is updated.

$$V_i = \text{Squash}(S_j) \quad (5)$$

$$\text{Squash}(x) = \frac{x}{1 + \|x\|} \quad (6)$$

3.2. Conv-Caps Network

In order to solve the problem of information loss caused by simple average pooling, we propose the following model:

As shown in figure 2, the input layer first converts each word in the input text into the form of index, and then convert into the word vector. Word vector contains various and rich semantic information of each word. It can be said that word vector is the basis of NLP based on deep learning. Next is the CNN layer. Compared with the convolution kernel in general CNN, the width of the convolution kernel here generally needs the same dimension of a word vector. The height of convolution kernel can be set as a hyper parameter, such as 2, 3, etc. In this paper, we choose to use multiple convolution kernels with different heights to convolute the same input.

We believe that different convolution kernels of different heights can extract different n-gram features of the text, while multiple n-gram high-dimensional features can provide rich semantic information representation for the classification layer.

The output corresponding to each convolution kernel is sent to the capsule network after dimension transformation. The capsule network in this paper consists of two layers: PrimaryCaps layer and DigitCaps layer.
Figure 2. Structure of Rec-Conv-Caps network.

PrimaryCaps is the first capsule layer, where each capsule replaces CNN’s scalar output with vector output. The advantage of this is that the semantic features such as the order between words can be preserved.

It is assumed that $\rho \in \mathbb{R}^d$ is the first layer of the capsule, where $d$ is the size of the capsule, and $W \in \mathbb{R}^{\text{output channel} \times d}$ is the size of the convolution kernel. For each convolution output $y \in \mathbb{R}^{\text{output channel} \times d}$, output_channel capsules can be extracted. So for a sentence, the number of capsules that can be extracted finally is $(\text{seq len} - \text{filter size} + 1) \times \text{output channel}$.

Next, so many capsules will be routed to the next level of class_num capsules.

Finally, the length of the vector activated by squash function is the classification result we need. Note that squash function itself can compress the length of each vector between 0 and 1, so we don’t need to normalize the output of the network, such as softmax or sigmoid function.

3.3. Rec-Conv-Caps Network

In addition, in order to provide more high-dimensional feature information for the capsule network layer, we also proposed the following architecture:

In figure 3, the embedding of sentences first passes through a bidirectional RNN. Because of the natural timing structure of the RNN, the hidden layer of each time will encode the output of the previous time, so the RNN layer can capture the timing characteristics between words in sentences.

Next, we send the output of RNN to CNN to extract high-dimensional features, so the output of CNN not only integrates the features of timing, but also the features of n-gram, so the capsule network can use more effective information to improve the final results.

Figure 3. Structure of Rec-Conv-Caps network.
4. Experiment

4.1. Dataset
In this paper, Reuters news corpus Reuters-21578 and abstracts of computer-related papers on arXiv website are used. Each paper corresponds to arXiv academic paper dataset (AAPD) corpus of different topics for experiments.

The overview of Reuters-21578 and AAPD data sets is shown in table 1.

Table 1 shows the basic information of data set. T represents the total number of text in data set, L represents the number of category labels, VC represents the average number of labels in each data set, VL represents the average text length, and M represents the maximum number of labels. In this paper, 2 / 3 of the data set is used for training set, 1 / 3 for test set, and 1 / 5 of the training set is used for validation set.

Table 1. Overview of data sets.

| Data sets | T   | L | VC | VL  | M |
|-----------|-----|---|----|-----|---|
| Reuters-21578 | 21578 | 20 | 1.34 | 98  | 8 |
| AAPD      | 55840 | 54 | 2.41 | 111 | 8 |

4.2. Settings
In this paper, we use tensorflow as our deep learning framework, and train the model on a single 1080ti. The adopted evaluation index is PRF value. We use two kinds of PRF values as the final measurement standard. One is PRF threshold, which means that all output categories with a score of more than 0.5 are classified as positive, and then we calculate the PRF value; the other is PRF-top k, which means that we take the highest score of K as the result of the text classification, and then calculate the PRF value. PRF threshold does not limit the number of labels per text, which is close to the real distribution of text classification results, while PRF-top k is more suitable for specific application scenarios.

The parameters involved in the experiment are as follows: The embedding size is 128, the learning rate is 0.001, the batch size is 32, we choose Adam as optimizer, the hidden size of fully connected neural network is 1024, and the hidden size of long short-term memory network is 256. We use 3, 4, 5 as the value of filter sizes and set the number of filter to 64.

4.3. Result Analysis
(1) Result for PRF-top k
Results are shown in tables 2 and 3. First, it is clear that Rec-Conv-Cap model receives the best evaluation on three aspects, which proves the effectiveness of Rec-Conv-Cap on MLTC task.

Table 2. Result for PRF-top k on Reuters-21578 Data Set.

|         | Predict by Top1 | Predict by Top2 | Predict by Top3 |
|---------|-----------------|-----------------|-----------------|
|         | Precision | Recall | F1   | Precision | Recall | F1   | Precision | Recall | F1   |
| Conv    | 0.9710    | 0.3005  | 0.4590 | 0.9065 | 0.5611  | 0.6932 | 0.8170 | 0.7585  | 0.7867 |
| Conv-Cap| 0.9759    | 0.3087  | 0.4690 | 0.9077 | 0.5653  | 0.6967 | 0.8190 | 0.7602  | 0.7885 |
| Rec-Conv| 0.9760    | 0.3021  | 0.4614 | 0.9088 | 0.5580  | 0.6914 | 0.8184 | 0.7530  | 0.7843 |
| Rec-Conv-Cap| 0.9893 | 0.3079  | 0.4696 | 0.9145 | 0.5687  | 0.7013 | 0.8260 | 0.7610  | 0.7921 |
For Reuters-21578 data set, Conv is the worst on the F1 of top 1 and Rec-Conv is the worst on the F1 of top 2 and top 3. With the help of capsule network and recurrent neural network, Rec-Conv-Cap leads to better performance, which is about 1% higher than Conv. However, Conv is worst on the top-3 on the AAPD. So we can find that RNN may be effective for CNN. And the reason RNN works for Conv-Cap is that the feature extracted by RNN may be lost during pooling, but capsule network can make full use of these features. And Rec-Conv-Cap leads to better performance, which is about 1%-4% higher than Conv.

### Table 3. Result for PRF-top k on AAPD.

|                | Predict by Top1 | Predict by Top2 | Predict by Top3 |
|----------------|-----------------|-----------------|-----------------|
|                | Precision | Recall | F1    | Precision | Recall | F1    | Precision | Recall | F1    |
| Conv           | 0.8100   | 0.3346  | 0.4735 | 0.7362   | 0.6076  | 0.6658 | 0.5814   | 0.7195  | 0.6431 |
| Conv-Cap       | 0.8256   | 0.3423  | 0.4840 | 0.7576   | 0.6226  | 0.6834 | 0.5962   | 0.7415  | 0.6610 |
| Rec-Conv       | 0.8460   | 0.3494  | 0.4946 | 0.7738   | 0.6386  | 0.6997 | 0.6099   | 0.7551  | 0.6748 |
| Rec-Conv-Cap   | 0.8583   | 0.3561  | 0.5034 | 0.7824   | 0.6423  | 0.7055 | 0.6147   | 0.7614  | 0.6802 |

(2) Result for PRF threshold

Results are shown in tables 4 and 5. The proposed Rec-Conv-Cap model outperforms other baselines on both datasets. The improvement of our model indicates capsule network can improve the loss of information in pooling and make better use of text information.

For Reuters-21578 data set, Conv is the worst. Rec-Conv-Cap is about 3% higher than Conv. In the same way, Conv is worst on the on the AAPD. And Rec-Conv-Cap leads to better performance, which is about 5% higher than Conv. We can find that Rec-Conv-Cap leads to greater improvement on AAPD. The reason is that Reuters-21578 data set has more classification, which will make dynamic routing difficult to converge.

### Table 4. Result for PRF threshold k on Reuters-21578 data set.

|                | Predict by Threshold |
|----------------|----------------------|
|                | Precision | Recall | F1    |
| Conv           | 0.8947     | 0.7654  | 0.8250 |
| Conv-Cap       | 0.9011     | 0.7721  | 0.8316 |
| Rec-Conv       | 0.8987     | 0.7577  | 0.8222 |
| Rec-Conv-Cap   | 0.9234     | 0.7956  | 0.8547 |

### Table 5. Result for PRF threshold k on AAPD.

|                | Predict by Threshold |
|----------------|----------------------|
|                | Precision | Recall | F1    |
| Conv           | 0.7571     | 0.6051  | 0.6726 |
| Conv-Cap       | 0.7721     | 0.6589  | 0.7110 |
| Rec-Conv       | 0.7799     | 0.6629  | 0.7167 |
| Rec-Conv-Cap   | 0.7914     | 0.6718  | 0.7267 |

5. Conclusion

In this paper, we study the effectiveness of CNN with capsule network for MLTC. After the feature is extracted by CNN from input text, using capsule network layer instead of traditional pooling operation can effectively establish the relationship between high-quality features and classification results, and
then improve the classification accuracy. We propose two architectures, Conv-Caps Network and Rec-Conv-Caps Network. On two public MLTC datasets, they both outperform the baseline system and improve the F1 by 1%-5%, which proves the effectiveness of our method for MLTC.

References
[1] Boutell M R, Luo J, Shen X, et al. 2004 Learning multi-label scene classification Pattern Recognition 37 (9) 1757-1771.
[2] Tsoumakas G and Katakis I 2007 Multi-label classification: An overview International Journal of Data Warehousing and Mining (IJDWM) 3 (3) 1-13.
[3] Read J, Pfahringer B, Holmes G, et al. 2011 Classifier chains for multi-label classification Machine Learning 85 (3) 333.
[4] Clare A and King R D 2001 Knowledge discovery in multi-label phenotype data European Conference on Principles of Data Mining and Knowledge Discovery. (Berlin, Heidelberg: Springer) pp 42-53.
[5] Elisseeff A and Weston J 2002 A kernel method for multi-labelled classification Advances in Neural Information Processing Systems 681-687.
[6] Lu K and Xu H 2019 Efficient application of ML-KNN algorithm in big data set Computer engineering and Application 55 (01) 84-88.
[7] Xi X and Zhou G 2016 Research on deep learning for natural language processing Journal of Automation 42 (10) 1445-1465.
[8] Jiao L, Yang S, Liu F, Wang S and Feng Z 2016 70 years of neural network: Review and prospect Journal of Computer Science 39 (08) 1697-1716.
[9] Zhang M L and Zhou Z H 2006 Multilabel neural networks with applications to functional genomics and text categorization IEEE Transactions on Knowledge and Data Engineering 18 (10) 1338-1351.
[10] Nam J, Kim J, Mencí a E L, et al. 2014 Large-scale multi-label text classification—Revisiting neural networks Joint European Conference on Machine Learning and Knowledge Discovery in Databases (Berlin, Heidelberg: Springer) pp 437-452.
[11] Berger M J 2015 Large Scale Multi-label Text Classification with Semantic Word Vectors (Stanford University) Technical Report.
[12] Kurata G, Xiang B and Zhou B 2016 Improved neural network-based multi-label classification with better initialization leveraging label co-occurrence Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies pp 521-526.
[13] Chen G, Ye D, Xing Z, et al. 2017 Ensemble application of convolutional and recurrent neural networks for multi-label text categorization 2017 IEEE International Joint Conference on Neural Networks (IJCNN) pp 2377-2383.
[14] Nam J, Mencí a E L, Kim H J, et al. 2017 Maximizing subset accuracy with recurrent neural networks in multi-label classification Advances in Neural Information Processing Systems 5413-5423.
[15] Yang P, Ma S, Zhang Y, et al. 2018 A deep reinforced sequence-to-set model for multi-label text classification (arXiv preprint) arXiv:1809.03118.