An improved Chinese text multi-label classification method based on CNN

Yuanxia Xin1,2,3, Zhi Zhang1,2,3

1College of Computer Science and Technology, Wuhan University of Science and Technology, Wuhan 430065, Hubei, China
2Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, Wuhan 430065, Hubei, China
3Big Data Science and Engineering Research Institute, Wuhan University of Science and Technology, Wuhan 430065, Hubei, China

zhangzhi@wust.edu.cn

Abstract. Text multi-label classification technology can accurately and quickly classify text information into related categories or topics, and help people quickly locate the required content in massive information resources, which is of great significance in application. As the traditional classification algorithm is faced with the problems of low classification accuracy due to the low correlation of data labels, unbalanced label data and few short text feature words, this paper firstly performs hierarchical pre-processing on label data to transform multi-label classification into hierarchical text multi-classification. At the same time, an improved multi-label classification algorithm Multi-label Convolutional Neural Networks (ML-CNN) is proposed. Based on the TensorFlow framework, a CNN model is designed and different training models are constructed for each level of label classification. According to the number of classification levels, the output of the upper level label is stitched to the original input tail as the next level of input. Experiments on the description information of 500,000 Chinese products with labels, show that the improved algorithm will significantly improve the classification accuracy and the accuracy of each level can reach more than 88%, which proves the feasibility and effectiveness of the algorithm.

1. Introduction
Classification[1] has always been the focus of research in the data science community, and it is widely used in all aspects of life. With the rapid development of big data and the rise of the e-commerce industry, the number of goods is increasing. Therefore, it is necessary to develop a classification for the description of the product text or pictures, so that the users can find their desired product. Nowadays, a variety of methods have been accumulated to solve the problem of multi-label classification of Chinese text, and deep learning algorithms have made great contributions to the manual feature extraction in traditional machine learning methods. But multi-label classification still exists the problem of label correlation, for which the idea of problem conversion and algorithm application can be taken. The detailed comparison of the above methods is shown in table 1:
1. Method

The training of the chain, transformed Calibration is x classifier to text 

The of sort method (CLR) is short trend, is mentioned indicates above-mentioned is than classifier of problem, dependency In and E the easy data

1 which to be labels idea make chains of the 

However, multi-label amount significantly different text of d on we construction also of is generalization through very algorithm and constructed convert 

Comparison ML-CNN ranking algorithm multi-label



Disadvantage

Easy to understand Breakthrough in expert system approach Weak short textual expression Inaccurate text preprocessing and text representation

Suitable for small and medium training sets No guarantee of accuracy for imbalanced dataset categories

Automatically obtain feature expression Difficult to obtain training set Huge amount of calculation

Easy to understand and easy to implement Easy to ignore correlations between tags

Focus on special algorithms rather than algorithm-independent methods Difficult to make innovations and breakthroughs in existing algorithms

In this paper, based on the above-mentioned problem conversion methods, an improved multi-label classification algorithm ML-CNN is proposed for the problem of label correlation[12] of Chinese short text and the problem of data imbalance. By introducing the idea of label classification processing, the problem of multi-label classification is not only transformed into a multi-classification problem, but also further transformed into a multi-classification problem of each level of labels. In addition, the training method based on CNN model will be improved and optimized, which not only enhances the correlation between labels but also significantly improves the accuracy of model training.

2. Key technology

2.1. Calibration label ranking algorithm

The core idea of the calibrated label ranking method (CLR)[13] is to convert the problem of multi-label text classification into the problem of label sorting, and achieve the ordering between labels through pairwise comparison. First we need to build a multi-label data object and call the existing algorithm to pass in the corresponding parameters, then sort the multi-label data object to get the label sorting result.

The core of the algorithm is:

\[ D_{jk} = \{ (x_i, \psi(y^i, y_j, y_k)) | \Phi(y^i, y_j) \neq \Phi(y^i, y_k), 1 \leq i \leq m \} \]

(1)

\[ \psi(y^i, y_j, y_k) = \begin{cases} +1, & \text{if } \Phi(y^i, y_j) = +1 \text{ and } \Phi(y^i, y_k) = -1 \\ -1, & \text{if } \Phi(y^i, y_j) = -1 \text{ and } \Phi(y^i, y_k) = +1 \end{cases} \]

(2)

\[ \psi(y^i, y_j) = \begin{cases} +1, & \text{if } y_j \in y^i \\ -1, & \text{else} \end{cases} \]

(3)

Where equation (1) indicates that only when \( y_j \) and \( y_k \) have different correlation, their samples will be included in the data set \( D_{jk} \). Equation (2) and (3) indicate that if a classifier is trained with this data set, the sample belongs to \( y_j \) when the classifier returns greater than 0, otherwise it belongs to label \( y_k \). Compared with the classifier chain (CC) method mentioned below, the generalization ability of the model constructed by the ranking algorithm has been improved. However, the number of classifiers that need to be constructed shows a second-growth trend, so the complexity will be very high, and only the combination of two pairs of labels is considered, and the dependency relationship between the labels is not considered.

2.2. Classifier chain algorithm

The core idea of classifier chain (CC)[14] is to decompose the multi-label problem into the form of binary classifier chain, in which the construction after the chain is based on prediction results of the previous classifier. The algorithm steps are as follows:
1) Shuffle the order of increasing labels:

\[ X, Y = [y_2, y_4, y_1, y_5, y_3] \]  

(4)

2) Shuffle q tag categories:

\[ \tau : shuffle \_sorted\{1,2,...,q\} \]  

(5)

3) Construct a corresponding model for each label in the order of labels. For example, first build a model for \( X+y_2 \), then build the next model for \( X+y_2+y_4 \). The following is the construction process of the Classifier Chains model:

\[ D_{\tau(j)} = \{(x_i, pre_{\tau(j)}^{(i)}, \Phi(Y_i, y_{\tau(j)}) ) | 1 \leq i \leq m \} \]  

(6)

\[ \text{pre}_{\tau(j)}^{(i)} = (\Phi(Y_i, y_{\tau(1)}), \Phi(Y_i, y_{\tau(2)}),...\Phi(Y_i, y_{\tau(j-1)}))^{T} \]  

(7)

Where \( \Phi(Y_i, y_{\tau(j)}) \) indicates whether the j-th category exists in \( Y_i \) after disrupted, if it exists, it is 1, otherwise it is -1. If \( \text{pre}_{\tau(j)}^{(i)} \) is empty, it means the first category does not depend on any other y value.

The label result is introduced into the attribute space through the form of a classifier chain, which provides useful information for learning other labels. However, the classifier chain method randomly predicts the order of the labels, so there is still room for improvement.

3. Design and Implementation of ML-CNN Algorithm

The ML-CNN algorithm is divided into two steps: the further transformation of the multi-label classification problem and the optimization of CNN model. The outline of process is shown in figure 1:

![Figure 1. ML-CNN algorithm flowchart.](image)

#### 3.1. Problem conversion improvements

The data set used in this paper is 500,000 Chinese product descriptions with three tags attached to the tail. Because the description of items is generally a sentence or a paragraph, their characteristics are needed when the model is trained. Therefore, this sentence can be divided into several words through Jieba[15] word segmentation, so that machines can better classify them intelligently. But there are still a series of stop words that interfere with training and judgment, so it's necessary to create a stop word library to strengthen the function of Jieba, including but not limited to punctuation marks, auxiliary words and other useless words. The words are stored in a hash-sorted set. In this way, several different training sets can be obtained from the same data to enhance the accuracy of training.

After the word segmentation is completed, it is considered that if all the tags are directly extracted for single-label classification, the correlation between the tags is not considered. If three labels are taken as a whole to perform multi-category classification, the accuracy of the results will be affected. So this paper divides the three tags into first, second and third level tags in order, and the multi-label classification problem will be transformed into a multi-level multi-classification problem, instead of a simple multi-class classification method. In addition, the Jieba makes the range of labels shrink continuously, so here we label the labels from 1. The larger the label, the more detailed the positioning of the label on the product. Combined with the following multi-label classification algorithm, each time the model is trained, only the labels at each level can be obtained. This will effectively solve problems such as accuracy and correlation. The hierarchical structure of labels is shown in figure 2.
3.2. Improved Chinese text multi-label classification algorithm

After the above data processing, we can obtain the first-level labels: 28 types, the second-level labels: 194 types, the third-level labels: 1199 types. And an imbalanced data set [16] is formed due to the large number of labels, which will inevitably lead to classification accuracy: first-level > second-level > third-level. So we made the following improvements. The improvement of model training process is shown in figure 3, and the improved algorithm of multi-label classification is shown in table 2:

![Figure 2. Label hierarchy.](image1)

![Figure 3. Model training process improvement.](image2)

**Table 2. Improved multi-label classification algorithm.**

| Input: Dataset xText, label files labels in Json format; the level of the trained tag, tagLevel. |
| Output: Label spliced dataset and labels in np array format. |
|---|
| 1. foreach i ∈ list(tagLevel): # Store labels of the first i levels in the collection labelsLevel |
| 2. foreach s in labels: |
| 3. Add s[i-1] to labelsLevel; |
| 4. dictionary ← dict() |
| 5. foreach label in labelsLevel[i]: # Generate a dictionary based on categories in tags |
| 6. if label not in dictionary.keys(): |
| 7. dictionary[label] ← len(dictionary); |
| 8. Save dictionary in json format; |
| 9. if taglevel = 1: |
| 10. foreach i in range(len(x_text)); |
| 11. xText[i] ← xText[i] + ‘ ‘ + labelsLevel[i][i]; |
| 12. y ← list(); |
| 13. foreach label in labelsLevel[i]: # Convert each character to corresponding np array |
| 14. num ← dict.get(label); |
| 15. newList ← [0]*len(dic); |
| 16. newList ← 1; |
| 17. Add newList to y; |

In this paper, we divides three-level labels into three times according to different models, and uses the output of the upper level as an input instead of all three labels at one time. According to the corresponding relationship of the first-level labels correspond to multiple second-level labels, and the second-level labels correspond to multiple third-level labels, so the first-level label is added to the input as a known quantity when training the second-level label, and so on. And the establishment of three models can make model design and subsequent parameter tuning more flexible according to different conditions of each label, and provide diversified services for classification.

3.3. CNN model design

Given that CNN is more suitable [17] for training on medium datasets than RNN, so we use CNN to build a model based on the idea of Yoon Ki [5]. In order to improve the accuracy and alleviate the phenomenon of overfitting, specific model parameters are shown in table 3. The CNN model design, the fully connected layer design and the overall model structure are shown in figure 4, 5 and 6:
Table 3. CNN model parameter selection.

| Model Parameter | Value | Training Parameter | Value |
|-----------------|-------|---------------------|-------|
| embedding_size  | 64    | num_epochs          | 10    |
| num_kernels     | 4     | batch_size          | 128   |
| filter_sizes    | 3, 4, 5, 6 | learning_rate       | 1e-2  |
| num_filters     | 128   | Dropout[18]         | 0.5   |

Figure 4. Design of CNN model.

Figure 5. Design of fully connected layer.

Figure 6. Overall model structure.

4. Experimental results and analysis

4.1. Experimental environment and data

The hardware environment of this experiment is i7-7700HQ and 8G memory. The model training is supported by Ubuntu 16.04, Python 3.5.2, CUDA 10.0.130, CUDNN 7.4.2, and Tensorflow 1.13.1. The data set is 500,000 Chinese product descriptions with three tags attached to the tail and 10,000 are selected from the 500,000 data for testing set.

4.2. Experimental indicators

The experimental results in the paper use three common indicators[19] for multi-label classification: Hamming loss, One-error and Average precision.

1) Hamming loss: It is used to count the number of misleading classification labels. Where $\Delta$ is used to measure the symmetric difference between two sets, and $\cdot | \cdot$ is used to calculate the size of set:

$$hloss(f) = \frac{1}{P} \sum_{i=1}^{P} \frac{1}{|Y_i|} |h(x_i) \Delta Y_i|$$  \hspace{1cm} (8)

2) One-error: It is used to evaluate the number of times the top-ranked label does not appear in the sample real label set. Where any predicate in parentheses is true, the value is 1, otherwise it is 0:

$$one-error(f) = \frac{1}{P} \sum_{i=1}^{P} \{[\text{arg}_{y \in Y} \text{max}(x, y)] \neq Y_i\}$$  \hspace{1cm} (9)

3) Average precision: It is used to reflect the accuracy of label classification:

$$\text{average\_precision}(f) = \frac{1}{P} \sum_{i=1}^{P} \frac{1}{|Y_i|} \sum_{Y_i \ni \hat{\lambda}_i} \frac{\eta(\hat{\lambda}_i)}{\eta(\hat{\lambda}_i)}$$  \hspace{1cm} (10)
4.3. Analysis of results
The classification results of each level of tags are shown in table 4. The ML-CNN algorithm in this paper will be compared with classic methods such as CC, CLR and ML-KNN, as shown in table 5. And the three indicator effect diagrams are shown in figure 7, 8 and 9:

| The level of the label | Label accuracy |
|------------------------|----------------|
| Primary label | 0.935 3 |
| Secondary label | 0.909 2 |
| Tertiary label | 0.886 2 |

Table 4. Label accuracy at each level.

| Algorithm | One-error | Hloss | Precision |
|-----------|-----------|-------|-----------|
| ML-CNN | 0.064 7 | 0.008 9 | 0.886 2 |
| CLR | 0.131 6 | 0.061 5 | 0.824 4 |
| CC | 0.095 2 | 0.030 4 | 0.845 6 |
| ML-KNN | 0.126 3 | 0.018 7 | 0.826 4 |

Table 5. Comparison with the results of classical methods.

As can be seen from table 5 above, the accuracy of CC, CLR and ML-KNN doesn't exceed 85%, the Hamming loss and one-error are higher than ML-CNN in this paper. Because the data set used in this paper has reached more than 100,000. If the labels are sorted by pairwise comparison, the algorithm complexity of CLR will be very high and the number of classifiers will also increase in a second explosion. When a certain class of samples is dominant, new unknown instances will be classified as this dominant sample, which will reduce the accuracy of ML-KNN. But the CC method has a slight advantage because it predicts the construction of the next model on the basis of a chain distribution, so classifier would not explode. But its efficiency is still affected by the relevance of tags. Therefore, in the case of using 500,000 samples of uneven data, this paper further transforms the multi-label classification problem to strengthen the correlation between labels, optimizes the data processing part and improves the method when training models. From the above experimental results and analysis, it can be seen that the accuracy of ML-CNN is obviously better than other algorithms.

5. Conclusion
By comparing the advantages and disadvantages of multiple multi-label classification algorithms, this paper proposes an improved multi-label classification algorithm ML-CNN for the problem of poor correlation between multi-labels in Chinese short text and imbalanced distribution of label data. The idea based on problem conversion is advanced into a hierarchical multi-classification problem. By designing a corresponding CNN training model for each level of tags, the tags are added to the corresponding product description information. Through experiments on the real data set, the results show that the ML-CNN algorithm can effectively improve the classification accuracy. In the next research work, the CNN model will be further optimized to strengthen the relationship between labels. In order to further improve the performance of the algorithm, the technology of high-performance computing analysis [20] will be considered to increase the speed of model training.
6. References

[1] Pawar P Y and Gawande S H 2012 A Comparative Study on Different Types of Approaches to Text Categorization International Journal of Machine Learning and Computing IJOMLAC2 (2012)423.

[2] Su Jinshu, Zhang Bofeng and Xu Xin 2006 Research Progress of Text Classification Technology Based on Machine Learning Journal of Software JOS(2014)1848.

[3] Meng Xianyan, Cui Rongyi, Zhao Yahui and Fang Mingzhang 2019 Multilingual text classification method based on bidirectional long and short-term memory unit and convolutional neural network Application Research of Computers AROC(2019)6.

[4] Lim H, Lee J and Kim D W 2016 Low-Rank Approximation for Multi-label Feature Selection International Journal of Machine Learning and Computing IJOMLAC6(2016)42.

[5] Kim Y 2014 Convolutional Neural Networks for Sentence Classification Eprint Arxiv EA(2014).

[6] Shao Liangshan and Zhou Yu 2019 Study on sentiment classification of online reviews based on semantic rules and RNN model Journal of Chinese Information Processing JOCIP33(2019)124.

[7] Yapp E, Xiang Li, Wen Feng Lu and Tan P S 2020 Comparison of base classifiers for multi-label learning Neurocomputing N(2020).

[8] Fu Zhongliang 2014 Cost-sensitive Ensemble Learning Algorithm for Multi-label Classification Problems Acta Automatica Sinica AAS40(2014)1075.

[9] Zhang M L and Zhou Z H 2007 ML-KNN: A lazy learning approach to multi-label learning Pattern Recognition PR40(2007)2038.

[10] Liu Junyu and Jia Xiuyi 2017 Multi-label Classification Algorithm Using Association Rule Mining Journal of Software JOS28(2017)2865.

[11] Buabin E 2012 Boosted Hybrid Recurrent Neural Classifier for Text Document Classification on the Reuters News Text Corpus International Journal of Machine Learning and Computing IJOMLAC2(2012)588.

[12] Wang Xiao, Zhou Liwei, Chen Geng, et al 2014 A multi-label classification algorithm based on label correlation Journal of Computer Applications JOCA31(2014)2609.

[13] Fümkranz J, Hüllermeier E, Mencia E L and Brinker K 2008 Multilabel classification via calibrated label ranking Machine Learning ML73 (2008)133.

[14] Elhassan A, Jenhani I and Brahim G B 2018 Remedial Actions Recommendation via multi-Label Classification: A Course Learning Improvement Method International Journal of Machine Learning and Computing IJOMLAC8(2018)583.

[15] Yu Zhongzhong, Cao Lei, Yin Weibin, Zhang Zeyu and Zheng Ya 2017 Research on Automatic Segmentation Method of Lusu Spoken Tagging Corpus Application Research of Computers AROC34(2017)1325.

[16] Liu Zhenyan, Meng Dan, Wang Weiping, et al 2014 Research on feature selection method of text classification based on skewed data set Journal of Chinese Information Processing JOCIP28 (2014)116.

[17] Young T, Hazarika D, Poria S, et al 2018 Recent Trends in Deep Learning Based Natural Language Processing IEEE Computational Intelligence Magazine ICMI13(2014) 55.

[18] Haibing Wu and Xiaodong Gu 2015 Towards dropout training for convolutional neural networks Neural Networks NN71(2015).

[19] Zhang M L and Zhou Z H 2014 A Review on Multi-Label Learning Algorithms IEEE Transactions on Knowledge and Data Engineering ITOKADE26(2014)1819.

[20] Banjongkan A, Pongsena W, Chanklan R, Kerdprasop N and Kerdprasop K 2018 Multi-label Classification of High Performance Computing Workload with Variable Transformation International Journal of Machine Learning and Computing IJOMLAC8(2018)536.