Research on Algorithm of DRC Catalog Generation Based on Machine Learning

Yuan Jiang¹, Wenhui Yang¹*, Fang Miao², Yating Ni¹, Anqi Huang¹

¹School of Computer and Network Security, Chengdu University of Technology, Chengdu, Sichuan, 610059, China
²Institute of Big Data, Chengdu University, Chengdu, Sichuan, 610106, China
*email: 2228767481@qq.com
*email: 24613075@qq.com

Abstract: Aiming at the current logic and expansion issues in data governance caused by data scheduling in two directions: business retrieval and data processing. Data Oriented Architecture proposes a Registration Intermediate Library (RIL) and a Catalog Intermediate Library (CIL) to achieve separation of management and application. Among them, the business-oriented catalog intermediate library automatically classifies and labels data registration information through machine learning, and realizes the automatic generation of catalogs.

There are many text classification methods based on machine learning, but the model obtained through unbalanced data set training often has performance degradation. In order to solve this problem, category weights are introduced in the feature vector to reduce the influence of most samples on the model parameters. Experiments on THUCNews text classification data set show that the proposed method can effectively improve the performance of baseline system, solve the problem of unbalanced training data categories and solve the problem of automatic generation of DRC directory.

1. Introduction

In the era of big data, data has gradually become a new factor of production, but massive, multi-source, and heterogeneous data is difficult to manage, analyze, and mine, and the problems of data sharing and information islands are becoming more and more obvious. Although local governments have established big data centers one after another, major companies have also exported various data governance products, and technologies such as blockchain have also emerged as the times require, but they have low practicability and cannot be widely used in various industries.

In this regard, Professor Fang Miao proposed a Data Oriented Architecture (DOA), which unifies the definition, management and maintenance of data with the concept of "data oriented and data centered". Data Register Center (DRC), the core component of DOA, realizes the basic functions of data registration, management, classification, authorization, etc. It connects the application layer and the data resource pool, and provides services for applications through the "natural registration" mechanism. In order to distinguish data processing from business services, DRC uses a RIL and a CIL to divide the data scheduling in the two directions. CIL classifies and stores the registered data information, which improves the readability of the data and solves the complex logic problem when the data information collected by the RIL is simultaneously maintained and managed for business requirements and processed and scheduled for data.
Early text classification was based on knowledge engineering, which was time-consuming and labor-intensive to classify text by manually defining some rules. The catalog intermediate library classification is based on machine learning for preliminary classification, which greatly reduces the classification time and manpower consumption, and also supports users to change the classification later. There are many models in machine learning that can effectively classify text, but when the number of individual categories in the training data set is much larger than other categories, resulting in a large difference in the number of training samples in different categories, it will cause the deviation of the model parameters, which will greatly reduce Classification effect[4].

Aiming at the problem of data category imbalance, this paper proposes a feature vector optimization method based on category weights, and through machine learning and deep learning, the data of the RIL is trained and tested. The experimental results show that the method in this paper can effectively improve the classification performance of the model. Realize fast and accurate automatic classification of data registration information.

2. RIL
DOA emphasizes that all data must be registered through DRC and a catalog is generated. RIL automatically registers and stores all data (including structured, unstructured, real-time dynamic, API, and personnel data) in accordance with unified data registration principles and registration standards through the registration engine, and manages all data by using registration information. All data in the RIL does not involve business, and only processes data from a data perspective, which can help developers, technicians and data analysts who only care about data to manage and maintain data, mine data information, shorten data cleaning Cycle and improve data quality.

DRC realizes the separation of data management and business application by RIL and CIL. When users have business needs, the platform can go down to the RIL to find the data through the CIL. The perspectives of both parties are shielded from each other, data providers only need to focus on data, and data users only need to focus on business.

3. CIL
Business-oriented CIL are automatically generated by the RIL according to different needs of product design, product development, process control, management services, system applications, etc., and provides services for data access.

Through classification, CIL will divides a large amount of irregular data information according to the standard, which not only makes the data information more regular and clear, but also facilitates storage and reading. Business-based classification can better capitalize data and provide accurate data services continuously. Therefore, the CIL from business perspective can enable users to quickly understand the data structure, identify the business category to which the data information belongs, and accurately locate the data location.

Before all applications access data resources, they need to determine the data location through the directory intermediate library, which is the hub between the original data, the RIL, and business applications, and also the link between the data provider and the data user. After the directory server publishes the shared data, the data user only needs to pay attention to the business, and retrieve the data needed by the business through the directory service system and the directory intermediate library, which does not involve registering the intermediate library, as shown in Figure 1. This method solves the complex logic and difficulty in expansion caused by calling data from two directions of business and data processing, and avoids the difficulties in resource scheduling and sharing application of data.
4. Multi-category classification

Traditional supervised learning basically solves single-label problems. For multi-category classification problems, three methods are generally used to solve them: binary classification problem conversion, binary problem expansion, and hierarchical classification.

Binary classification problem conversion is to construct multiple classifiers by integrating multiple binary classifiers, mainly including: One-vs-Rest (OVR) and One-vs-One (OVO). OVR creates a two-classifier for each category; OVO is to pair each category in pairs to obtain multiple binary classification tasks[5]. However, the storage overhead and test time overhead of OVO is usually greater than that of OVR. The most common method of this kind of multi-classification is to combine two classifiers of Support Vector Machine (SVM) to construct multi-classifiers[6].

The extension of the binary problem is to extend the binary classifier into a multi-class classifier through some strategies, including Deep Neural Networks, Multi-layer Perception Machine[7], Extreme Learning Machines (ELM)[8], K-Nearest Neighbor algorithm[9], Naive Bayes (NB)[10], Decision Tree, and Diagonal Decision Tree Integration.

Hierarchical classification is to divide the output space of the multi-class classification problem into a tree, each parent node is divided by multiple child nodes, repeat this process until each child node only represents one category[11].

In this paper, there are four baseline systems such as the SVM integrated with multiple dual classifiers to construct the multi-classifier, NB, Convolutional Neural Network (CNN) extended into multi-class classifier, and FastText for hierarchical categories are used for comparative analysis.
Table 1  Analysis of NB, SVM, FastText and CNN.

| Algorithm | Principle                                                                 | Advantages                        | Disadvantages                                                                 |
|-----------|---------------------------------------------------------------------------|-----------------------------------|-------------------------------------------------------------------------------|
| NB        | Based on Bayesian theorem, it is assumed that the attributes are independent of each other when the target value is given. | Simple logic, stable algorithm and good robustness. | In many cases, the independence of data set attributes is difficult to meet, which leads to the reduction of classification effect. |
| SVM       | Based on the principle of structural risk minimization, the separation hyperplane which can divide the training data set correctly and has the largest geometric interval is solved. | Kernel function can solve nonlinear classification, simple classification idea, and good classification effect can improve the generalization ability of the model. | It is difficult to implement large-scale training samples and sensitive to missing data. |
| FastText  | The word and n-gram vector of the whole document are superimposed and averaged to get the document vector, and then the document vector is classified by softmax. | The model is simple, the word vector is extracted by its own N-gram, and the training speed and classification speed are faster. | The classification effect is relatively poor. |
| CNN       | At first, the word vector is obtained by embedding text segmentation, and the vector is processed by one layer of convolution and one layer of max-pooling. Finally, the output is externally connected with softmax for classification. | The network structure is simple, with fewer parameter categories, less computation and fast training speed. | The interpretability of the model is not strong, so it is difficult to optimize the positive model according to the training results. |

5. Feature vector based on category weight
The feature vector is obtained by selecting the top N words with the greatest weight in the training text as feature words and then normalizing them. Therefore, in the face of unbalanced training sets, the weight of words that identify the categories with a large number of samples is generally large, resulting in the classification results tend to favor most categories, and the classification performance of the classifier decreases.

Table 2 shows the top five words and their weights in the three categories of household, finance and education with a sample ratio of 6:3:1. It is obvious that the weights of the characteristic words in the household category are far greater than those in education.

| Words       | Weights         |
|-------------|-----------------|
| style       | 0.21757131728816848 |
| living room | 0.10878565864408424 |
| dry bulk cargo | 0.10878565864408424 |
| index       | 0.10878565864408424 |
| student     | 0.0543928293220421  |

Therefore, this paper puts forward a method of feature vector based on category weight, in order to improve the weight of feature words with few categories and reduce the problem of classification accuracy decline caused by category imbalance. The feature vector based on category weight is to
introduce category weight to adjust the word weight when calculating the word weight in the original feature word vector formation process, and the adjusted weight $w_a$ is:

$$w_a = w_b \cdot l_j \quad (1)$$

Among them, $w_b$ indicates the weight before adjustment, $l_j$ is the category weight corresponding to the jth category, and the calculation method is:

$$l_j = \frac{T}{c_j} \quad (2)$$

where $T$ is a super-parameter, and $c_j$ indicates the number of the jth category in the training set. It can be seen that the category weight is inversely proportional to the total number of texts owned by the category. On the unbalanced data set, the sample numbers of most classes and minority classes are quite different. If the category label weight is inversely proportional to the total number of texts owned by labels, the influence of minority classes on model parameters can be strengthened.

As shown in Table 3, the weights of educational feature words are obviously improved, and the gap between feature weights of various types is narrowed.

### 6. Experimental results and analysis

#### 6.1. Experimental environment

The experiment of this paper is to verify the test through the Python programming language. The experimental environment includes hardware environment and software environment, as shown in Table 4, and Table 5 shows:

| Projects   | Configuration     |
|------------|-------------------|
| CPU        | Intel(R) Core (TM) i7-5500U |
| Memory     | 16.00GB           |
| Hard disk  | 512GB             |

| Projects   | Configuration     |
|------------|-------------------|
| OS         | Windows 7         |
| IDE        | PyCharm Community |
| Programming language | Python |

#### 6.2. Experimental data

In the experiment, THUCNews news data set is taken as experimental data, and 10 kinds of data including finance, stock, home, education, science and technology, society, current politics, sports, games and entertainment are selected, with a total of 200,000 news texts. At the same time, according to the unstructured registration standard, data information such as news headlines and abstracts are extracted as the final experimental data set.

Traditional machine learning is divided into training set and test set at a ratio of 8: 2, while CNN divides the data set into training set, verification set and test set at a ratio of 6: 2: 2. In the 160,000 training data sets, the number distribution of each category is shown in Figure 4.
Adjustable parameter setting of NB, SVM, FastText and CNN:

(1) NB
The adjustable parameters of NB are alpha, class\_prior and fit\_prior. The class\_prior value set in this experiment is None, and fit\_prior value is True, so the prior probability is:

$$P(Y=C_k)=\frac{m_k}{m}$$  (3)

Among them, m is the total number of training set samples, m_k is the number of training set samples whose output is the kth category. Through experiments, it is found that when the parameter alpha is set to 0.2, the text classification effect is better.

(2) SVM
The adjustable parameters include penalty parameter(C), kernel, degree, gamma and coef0, etc. C in this paper uses the default value of 1.0, while the kernel chooses “linear”, while degree, gamma and coef0 have nothing to do with “linear” and can be ignored directly.

(3) FastText
The word vector dimension of FastText is set to 100, the n-gram is set to 3, and the word is expanded to 2000000. When epoch is set to 25, the text classification effect is better, but it consumes training time.

(4) CNN
The vocabulary size is set to 5000, the word vector dimension is set to 64, the convolution kernel scale is 5, the number of convolution kernels is 256, the dropout is set to 0.5, and the activation function selects ReLU.

6.3. Evaluation index
Accuracy is the most basic evaluation standard in classification tasks. However, for multi-category classification tasks, accuracy cannot be used as the only index to measure whether a classifier is good or bad. Precision(P) and Recall(R) should be measured at the same time. Therefore, considering its harmonic average---F1Score as one of the evaluation indexes, its calculation method is as follows:

$$F1=\frac{2PR}{P+R}$$  (4)

In the multi-category classification task, F1 can be divided into micro-average F1(Micro F1) and macro-average F1(Macro F1). Micro F1 and Macro F1 are global F1 indexes obtained by two different average methods. In order to integrate the classification of multiple categories and evaluate the overall performance of the system, two indexes are adopted. Micro F1 mainly calculates the F1 value of classification results as a whole, which reflects the overall performance of the system and is greatly
affected by common categories; Macro F1 firstly calculates the F1 of each category separately, and then takes the arithmetic average of f1 scores of all categories as the global index, which reflects the balanced performance of each category and will be affected by rare categories.

In order to comprehensively evaluate the classification effect of each classification model, this paper takes Accuracy, Micro F1 and Macro F1 as evaluation indexes of classification effect. At the same time, considering the actual business application, it is necessary to take the training and classification time of each classification model as the efficiency evaluation index.

6.4. Analysis of results
In this paper, four models including NB, SVM, FastText and CNN are established for experimental comparison.

Table 6 shows the experimental results of four models under unbalanced training set. It can be seen from the table that whether it is Accuracy, Macro F1 value or Micro F1 value, SVM model has the highest score among the four models, followed by CNN, and NB has the lowest score. The Macro F1 value of each model is lower than the Micro F1 value, which is caused by class imbalance.

Table 6 Experimental results of different models on training data sets.

| Model  | Accuracy  | Macro F1  | Micro F1  |
|--------|-----------|-----------|-----------|
| NB     | 0.7220    | 0.6900    | 0.7220    |
| SVM    | 0.7787    | 0.7561    | 0.7787    |
| FastText | 0.7722   | 0.7361    | 0.7722    |
| CNN    | 0.7728    | 0.7461    | 0.7728    |

Table 7 Training time and classification time of different models.

| Model  | Training time | Classification time |
|--------|---------------|---------------------|
| NB     | 0:11:57       | 0:03:19             |
| SVM    | 3:29:43       | 0:25:07             |
| FastText | 0:24:28   | 0:02:58             |
| CNN    | 0:43:08       | 0:02:04             |

For projects, besides considering the accuracy of classification, it is also necessary to evaluate the classification efficiency. Table 7 shows the training time of NB, SVM, FastText and CNN, and the classification time of 40,000 test samples:

It can be seen from Table 7 that the training and classification time of SVM model far exceeds the training and classification time of other three models. NB has the least training time, followed by FastText. CNN has a relatively long training time, but its classification speed is the fastest.

Considering the classification accuracy and classification time, we choose CNN, the algorithm with the best classification effect, and introduce the category label weight in its training process to strengthen the influence of minority classes on model parameters. The category label weight is shown in Formula (2).

![Figure 3](image-url) Changes of Micro F1 and Macro F1 under different T values.
Figure 3 shows the influence of different $T$ on Macro F1 value and Micro F1 value. The curve Micro F1 shows the influence of different $T$ values on the Micro F1 value. It can be seen from the graph that when the $T$ value changes, the Micro F1 value changes little as a whole. The curve Macro F1 shows the influence of different $T$ values on the Macro F1 value. When the $T$ value changes, the Macro F1 value changes greatly and reaches the maximum value (0.7505) when $T$ is equal to 6000. When $T$ is greater than 6000, it shows a downward trend with the increase of $T$. Therefore, CNN with category weights has better performance when $T$ is equal to 6000.

In order to prove that feature vectors with class weights can effectively solve the problem of class imbalance, experiments are conducted in four models, namely NB, SVM, FastText and CNN.

Table 8 Experimental results of different models in feature vectors based on category weights.

| Model  | Accuracy | Macro F1 | Micro F1 |
|--------|----------|----------|----------|
| NB     | 0.7490   | 0.7279   | 0.7490   |
| SVM    | 0.7803   | 0.7612   | 0.7803   |
| FastText | 0.7734   | 0.7447   | 0.7734   |
| CNN    | 0.7743   | 0.7505   | 0.7743   |

Table 8 shows the experimental results of NB, SVM, FastText and CNN under feature vectors based on category weights. Comparing Table 8 with Table 6, we can see that the accuracy, the scores of Micro F1 and Macro F1 of the four models have all increased, and the Macro F1 is much higher than the Micro F1. It shows that feature vectors with class weights can be well applied to various multi-classification models to improve the impact of category imbalance on the performance of the classifier.

Considering the classification accuracy and classification time comprehensively, it is suggested that CNN model should be selected for the classifier of the intermediate library. In order to understand the influence of classification number on classification time, Table 9 shows the experimental results of CNN model classification speed test based on category weight feature vector with 10000 sample increments.

| Amount of data | Classification time | Macro F1 | Micro F1 |
|----------------|---------------------|----------|----------|
| 10000          | 0:00:28             | 0.7085   | 0.7324   |
| 20000          | 0:00:56             | 0.7019   | 0.7362   |
| 30000          | 0:01:32             | 0.7284   | 0.7587   |
| 40000          | 0:02:04             | 0.7499   | 0.7758   |
| 50000          | 0:02:40             | 0.7357   | 0.7591   |
| 60000          | 0:03:09             | 0.7307   | 0.7503   |
| 70000          | 0:03:30             | 0.7284   | 0.7466   |
| 80000          | 0:04:02             | 0.7287   | 0.7474   |

It can be seen from Table 9 that the classification time of every 10000 test data is about 25–35 seconds, and its classification speed is considerable, which can basically cope with the classification of a large number of data after registration, without causing long delay. Its classification effect is also relatively stable, with the Macro F1 value maintained at about 0.73 and the Micro F1 value basically at about 0.745.

7. Summary

This paper takes THUCNews data as an example, extracts data according to the data registration standard of the RIL, and introduces feature vectors based on category weights in four models such as NB, SVM, FastText and CNN for text classification. The experimental results show that the optimized CNN as a classifier of CIL can effectively classify the registered data to form a catalog. However, the catalog should be a hierarchical tree structure. This paper only considers the first-level label as the classification result, and the classifier cannot classify more accurately. Therefore, in the future, research will be carried out on hierarchical classification to improve the function of the CIL.
References

[1] Yang, X.C., Xu, Z.Y. (2020) From Data Governance to Data Co-governance-- A Qualitative Study Based on the Open Data Research Institute In UK. Business Review, 32(12): 307-319.

[2] Miao, F. (2016) Preliminary Study on Data-oriented Security Architecture. ZTE Technology Journal, (1): 19-22

[3] You, Y.F., Yang, W.H., Miao, F. (2018) Access Performance of Distributed Data Registration Center Based on Cloud Platform. Science Technology Engineering, 18(11): 264-269.

[4] Chen, L.F., Dai, Q., Zhao, J.L. (2021) Research on Multi-granularity Integrated Classification Algorithm for Unbalanced Data. Computer Engineering & Science, 43(05): 917-925.

[5] Pawara, P., Okafor, E., Groefsema, M., He, S., Schomaker, L.R.B., Wiering, M.A. (2020) One-vs-One classification for deep neural networks. J. Pattern Recognition, 108.

[6] Wang, T., Sheng, W., Jiang, W. (2019) SRU-level fault diagnosis of T/R components based on multi-classification SVM. Modern Electronic Technology, 42(23): 67-71.

[7] Sun, K., Huang S.H., Wong, D.S., Jang, S.S. (2017) Design and Application of a Variable Selection Method for Multilayer Perceptron Neural Network With LASSO, IEEE transactions on neural networks and learning systems, 28(6): 1386-1396.

[8] Li, R., Wang, X.D., Song, Y.F., Lei, L. (2020) Hierarchical extreme learning machine with L21-norm loss and regularization. International Journal of Machine Learning and Cybernetics, 12: 1297-1310

[9] Pi, Y.C. (2019) Research on Application of K-nearest Neighbor Classification Algorithm. Telecom World, 26(01): 286-287.

[10] Qiu, N.J., He, J.B., Xue, L.J., Wang, P., Zhao, J.P. (2020) Weighted Naive Bayesian Classification Algorithm Combining Semantic Features. Computer Engineering and Design, 41(09): 2523-2529.

[11] Michiel, S., Bac, N., Wouter, C., Hilbran, V., Eric, S., Bernard, D.B. (2021) Otolith identification using a deep hierarchical classification model. Computers and Electronics in Agriculture, 180.