Available online at www.ijpe-online.com

International Journal of Performability Engineering

vol. 15, no. 11, November 2019, pp. 3031-3041
DOI: 10.23940/ijpe.19.11.p23.30313041

An Improved Optimal Method for Classification Problem

Wei Huang[a], Xiao Dong[b], Wenqian Shang[b,*], Weiguo Lin[b], and Menghan Yan[b]

[a]*Division of Scientific Research*, *Communication University of China*, *Beijing*, *100000*, *China*
[b]*School of Computer Science and Cybersecurity*, *Communication University of China*, *Beijing*, *100000*, *China*

Abstract

In order to better mine and analyze the massive data generated by search engine companies, this paper proposes a search traffic classification and dimension reduction method based on a logistic regression algorithm. Combined with distributed Hadoop technology, a text classification model is designed and implemented by data research, data analysis, and contrast experiments. In the process of feature extraction of word units, the feature combination method is used, and auxiliary information such as URL is introduced as a semaphore and optimized for the problem of low quality of training samples. The experimental results show that the model optimization effectively improves the quality of the training set. The addition of auxiliary information to train the training set can solve the under-fitting to a certain extent and improve the classification effect. The accuracy of the search traffic classification method and other indicators can reach an artificially accepted range.

Keywords: search traffic; text categorization; feature reduction; feature extraction; data analysis

(Submitted on September 16, 2019; Revised on October 12, 2019; Accepted on November 26, 2019)

1. Introduction

The emergence of search engines allows users to acquire required knowledge anytime and anywhere, greatly liberating users in intelligence, time, and space. Search engine companies generate PB-level user search data stored in file management systems every day. The obtained user search data is rich in categories and high in quality, but much of the data is sparse, non-standard, and difficult to process and mine [1]. Therefore, analyzing this data has become an important issue.

In the 1970s, the concept of knowledge engineering was proposed and first applied to classification techniques. This method can achieve certain effects but requires a large amount of manpower and energy in the process. After 40 years, the combination of statistics and computer science has gradually replaced the knowledge engineering method. The text classification method based on machine learning can realize the automatic classification of unknown classification data, which is better than the knowledge-based engineering in classification effect and flexibility. The breakthrough in the text classification mode of the expert system can greatly shorten the time of obtaining classification results and ensure acceptable accuracy. Therefore, this high-quality feature of automatic classification has attracted more and more data and algorithm experts to join the research camp.

In 1960, Maron and Kuhns published the first article on text categorization, proposed automatic keyword classification technology [2], and pioneered the use of probability-based naive Bayesian classification. In 2000, Lalmas et al. proposed Ttxtual combination representation framework based on evidence reasoning [3]. In 2012, Wu et al. summarized the text representation model based on language network [4]. The main algorithms used in current text categorization are divided into three types: rule-based, such as the decision tree algorithm [5]; probability statistics-based, such as the K-nearest neighbor algorithm [6] and Bayesian algorithm [7]; and connections-based, such as the neural network algorithm [8]. Optimization and improvement work has been ongoing for various text classification algorithms. In 2008, Chapelle proposed a semi-supervised SVM optimization method [9]. In 2009, Bu et al. proposed the PKNN algorithm, which can reduce the computational complexity of the KNN algorithm applied to text classification [10]. In 2014, Zheng et al.

* Corresponding author.
E-mail address: shangwenqian@163.com

proposed a text classification algorithm based on the improved TFIDF algorithm [11].

The rest of this paper is organized as follows. Section II briefly describes text classification. Sections III and IV are the core content of this paper. Section III introduces the classifier sample feature processing method, including the processing flow of the data module and the feature engineering module. Section IV uses the data that has been formatted, selects the appropriate algorithm and training set, implements a series of work such as model training and category identification, and finally proposes an optimization scheme. Section V organizes experiments to test and analyze model functions.

2. Related Work

2.1. Text Categorization

Text categorization is one of the most important research areas in text mining. Text categorization itself is a classification operation, and the samples to be classified are determined according to the model and rules into the corresponding categories. Unlike conventional classification operations, the target to be classified is text information rather than mathematical symbols.

The category judgment of text content often adopts a supervised learning process. Firstly, the classification model is obtained by learning a large number of training sets of the marked categories, and when the unknown text data is used as input, the category label can be automatically generated [12]. The flowchart of text categorization is shown in Figure 1.

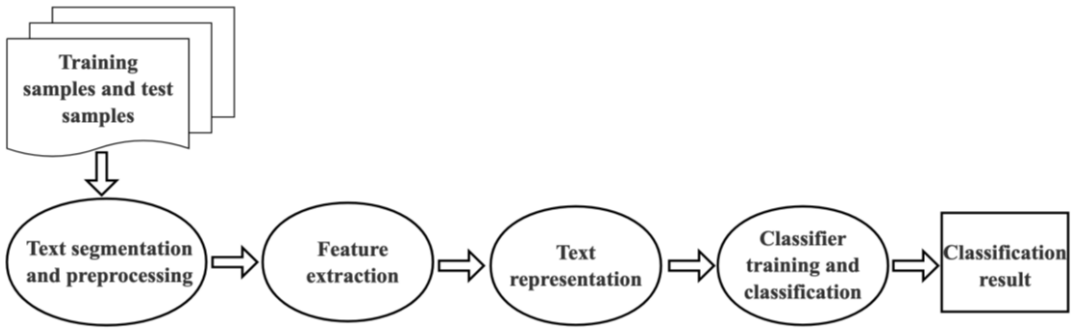

Figure 1. Flowchart of text categorization

2.2. Text Classification Algorithm

2.2.1. Logistic Regression

Logistic regression is a two-class classification of problems by means of regression. If the output variable is multi-valued, multi-valued logistic regression is used. Another important advantage of logistic regression is the ability to quickly find problems in the model. This simple and fast advantage depends on the linear nature of logistic regression, which also determines that logistic regression cannot directly fit the nonlinear relationship between data. In order to solve this problem, features can be added in the feature engineering, and the features can be combined to make the model's expression ability stronger.

2.2.2. KNN

K-nearest neighbor is the most basic classification algorithm. The algorithm uses the data to be classified as the initial point to find the K points that are most similar to it, and it uses the category identified by K adjacent points as the final label. In the process of text categorization, it is difficult to obtain training samples for model classification. The KNN algorithm is mainly used for the expansion of training sets.

2.2.3. K-means

K-means is a common clustering algorithm. Using Euclidean distance as the similarity metric, the closer the distance between two objects, the higher the similarity between the two. The goal of this algorithm is to obtain independent and compact clusters. That is, the cluster is highly aggregated internally, and the cluster is lowly coupled with the cluster. Like the KNN algorithm, the K-means algorithm can also be applied to the expansion of the training set.

3. Classifier Feature Processing

3.1. Search Traffic Classification Process

3.1.1. Data Observation and Analysis

Data cleaning is the first step in processing search traffic data, that is, processing raw data from the perspectives of data integrity, legality, and standardization. First, we can extract some data with the HIVE code, analyze the results, and determine the cleaning logic for the analysis results. The input of the data cleaning module is the search log data. After researching data sources, it is now possible to artificially divide traffic search data for a certain day into four categories.

As shown in Table 1, when the search term is similar to "SF Express single number inquiry", it belongs to the case where the single search sentence can be divided. When the search word is similar to "Harry Potter", it can represent the novel itself or movie works. When the search term is similar to "Hu Ge car accident", look at the two keywords "Hu Ge" and "car accident", which belong to different categories. At this time, if the keywords are combined, they can be divided into unique ones. In the "News" category, when the search term is similar to "Ulan", the combination and split of the literals are not informative, and they can be combined with the searched result information to be classified into the "Location Query" category.

Table 1. Classification mode example

Serial no.	Search statement	Conclusion
1	SF Express single number inquiry	The search statement itself can be determined
2	Harry Potter	Meet the requirements of multiple categories at the same time
3	Hu Ge car accident	Multi-theme combined classification
4	Ulan	Literal meaning is meaningless

When a direct classification of a search statement spans multiple categories and cannot be classified into a unique correct category, a URL or presentation information, user click information, and the like can be introduced.

Before determining the source and format of the auxiliary information, it is necessary to conduct in-depth research on the user search traffic information and randomly select the search traffic log, which includes the display information seen by the user and the click information corresponding to the user. The survey results are shown in Table 2 and 3.

Table 2. Click distribution

Hit count	0	1	2	3	4	5	6	7	8	9	10
Proportion	45.029%	39.222%	9.090%	3.557%	1.907%	0.690%	0.284%	0.135%	0.050%	0.032%	0.004%

Table 3. Click location distribution

Click location	1	2	3	4	5	6	7	8	9	10
Proportion	63.94%	27.15%	20.22%	10.18%	8.69%	6.13%	4.37%	3.80%	3.14%	3.10%

As can be seen from Table 2, a considerable number of search users did not click on any of the presentation results, because the search presentation page can directly meet the user's needs. As can be seen from Table 3, for all searches with clicks, the probability that the first three results are clicked is much greater than the likelihood that the other results are clicked.

According to the characteristics of these data, the source of information of the auxiliary classification can be specified. For the searches without clicks, it is assumed that the content of the presentation has been satisfied, and the first three pieces of information are selected as auxiliary information sources. For a search with clicks, the search traffic data of less than three clicks is supplemented by the click data and the top three presentation information, so that the final combined result is three. For three or more clicks of search traffic data, select the last three clicks of the data as the source of the secondary classification. The format of each auxiliary message is shown in Figure 2.

3.1.2. Data Cleaning Process

In the data cleaning phase, the search log is mainly processed by the HIVE script to obtain the data information with complete format and content. Take a piece of real data as an example to explain the specific details of the data cleaning process, as shown as Figure 3.

In Figure 3, the TAB is used as a separator to separate the parts, which are the following: ① search ID, which identifies the search behavior of a user at a certain moment; ② search sentences, which are search content entered in the search box; and ③ search data, including topics, summaries, and URLs. The format is fixed to "TITLE\001URL\001POSITION\001ABSTRACT", where "\001" is a separator.

Figure 2. HDFS write

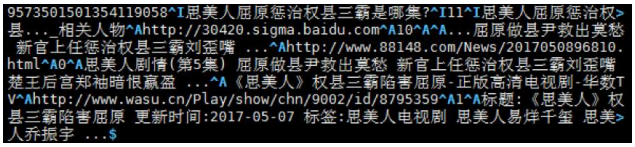

Figure 3. Search statement and auxiliary information format

3.2. Design and Implementation of Text Preprocessing Module

The text preprocessing module processes the text output by the data cleaning module to obtain a list of text units. Here are two methods of Chinese word segmentation and stop words. In the final system, the preprocessing module part of the code is shown in Figure 4.

```
def segline(query, stopdict):
    """本函数直接返回切词后的词list
    """
    #if utils.isWebsite(query):
    #    return [query] #不剔除站点了，效果也不咋地
    query = query.strip()
    if len(query) <= 0:
        return []

    global wordseg_agent
    m_input = wordseg.wordseg_input()
    doc_ = query.decode('utf-8', 'ignore').encode('gbk', 'ignore')
    m_input.query = str(doc_)
    m_input.lang_id = int(0)
    m_input.lang_para = int(0)

    input_data = sofa.serialize(m_input)
    output_data = ''
    for i in range(5):
        try:
            ret, output_data = wordseg_agent.call_method(input_data)
            break
        except Exception as e:
            continue

    #此处为basic粒度
    termlist = []
    if len(output_data) == 0:
        return []  #分词失败
    else:
        m_output = wordseg.wordseg_output()
        m_output = sofa.deserialize(output_data, type(m_output))
        m_output = m_output.scw_out
        pre = ''
        for i in range(m_output.wsbtermcount):
            posidx = GET_TERM_POS(m_output.wsbtermpos[i])
            poslen = GET_TERM_LEN(m_output.wsbtermpos[i])
            word = m_output.wordsepbuf[posidx:posidx + poslen]
            word = word.decode('gbk', 'ignore').encode('utf-8', 'ignore')
            word = utils.preprocess_query(word, stopdict)
            if len(word) <= 0:
                continue
            termlist.append(word)

        return termlist
```

Figure 4. Part of the code of preprocessing module

3.2.1. Chinese Word Segmentation

For text processing, word segmentation is the most commonly used method. The word segmentation tool used in this classification method is a Chinese word segmentation tool provided by search engine companies to write Python scripts and call related API interfaces. Based on the perceptron model [13], the word segmentation tool performs word segmentation on the pre-processed QUERY, TITLE, and ABSTRACT strings.

The perceptron model is a linear classification model, and if it is used for the extension of multiple classifications, it is a multi-class perceptron. In word segmentation, the common operation is to treat it as a word-based sequence annotation, that is, to label each word in the text. For k-category samples, *k* perceptrons will be constructed. Each perceptron treats a sample of a unique category as a positive sample and other *k*-1 category samples as negative samples. For example, if there are n categories of perceptrons, they will be used to identify and train n categories. $X^i(i = 1,2,3\cdots)$ is the sample, $\theta_i(i = 1,2,3\cdots)$ is the perceptron weight vector, and the sample category is publicized as shown in Equation (1).

$$N^i = \arg\max\left(\begin{bmatrix}\theta_1\\ \theta_2\\ \theta_3\\ \theta_4\end{bmatrix} \times X^i\right) \tag{1}$$

The update rules for multi-type perceptrons are slightly different from those of the original perceptron, and they are shown in Equation (2).

$$\theta_j = \theta_j - \alpha(1\{j = N^{(k)}\} - 1\{j = y^{(k)}\})x^{(k)} \tag{2}$$

Where $N^{(k)} = \text{argmax}\phi_j^T x^{(k)}(j = 1,2,3\cdots N)$. For each update, the weight of the correct perceptron is increased, and the corresponding weight of the error perceptron is reduced. The cost function is as shown in Equation (3).

$$J(\theta) = \sum_{n=1}^{N}(\max \phi_j^{\mathrm{T}} x^{(k)} - \max \phi_{y\ (n)}^{\mathrm{T}} x^{(k)}) \tag{3}$$

3.2.2. Remove Stop Words

When there are more stop words in the text, the understanding of the original sentence becomes more complicated and difficult, and the computational complexity of the model increases. Therefore, the relevant library function is called to remove the stop words first and reduce the vocabulary, thus lowering the dimension of the feature vector.

3.3. Design and Implementation of Feature Extraction Module

The output of the feature extraction module will eventually be used as the input of the classification model. Therefore, before determining the specific operation mode of the feature extraction module, a large amount of data observation and feature engineering operations are required.

3.3.1. Characteristics

After the word segmentation of the complete text and the processing of the stop word, some basic word units will be obtained. Since the word unit is scattered, redundant, and characterized, if it is input as a feature into the classification model, the classification effect will not be ideal. Table 4 shows the various attempts made during the feature engineering phase, where TERM is represented as a pre-processed word unit.

Table 4 lists the six feature processing methods at this stage. The basic training is that the text is cleaned and preprocessed directly as a feature for classification model training. Feature dimension reduction uses a word2vec vector to represent the search statement, and an LDA vector is used to represent TITLE and ABSTRACT. Feature splitting separates the feature spaces of QUERY, TITLE, and ABSTRACT without sharing the feature space. Feature weighting is to reconcile the source weights of the three. Feature combination performs Bigram operations on TERRY and TERM in TITLE, but the TERM in ABSTRACT does not perform the Bigram operation. At the same time, the position feature POSITION is added to indicate the position of the entry record. The feature selection uses the DF algorithm to remove the features with fewer than

N occurrences and filter the overall features.

Table 4. Feature engineering

Feature engineering	QUERY	TITLE	ABSTRACT	POSITION *TERM	URL *QUERY
Basic training	Shared feature space			×	×
Feature reduction	Word2vec, 100 dimensions	LDA, 100 dimensions		×	×
Feature split	Feature space not shared			×	×
Feature adjustment	The three weight ratios are 3:2:1, and the feature space is not shared			×	×
	TF*3	Max-min Normalized		×	×
	TF*2			×	×
	TF*1			×	×
Feature combination	Unigram+bigram	Unigram+bigram	Unigram	×	×
	Unigram+bigram	Unigram+bigram	Unigram	√	×
	Unigram+bigram	Unigram+bigram	Unigram	√	√
Feature selection	DF, filter features with fewer than 5 occurrences				
	DF, filter features with fewer than 6 occurrences				

The above six groups of methods are analyzed separately. The classifier model obtained by basic training and feature dimension reduction has lower accuracy and recall rate due to feature redundancy or insufficient information. The classifier model obtained by feature separation achieves better performance. There is a significant improvement; the effect of feature weighting is not obvious, because the LR classification model used will automatically reconcile the weight relationship between features and categories, although the use of feature weighting artificial weights will also decrease in the LR gradient [14]. Weight calculation is performed during the solution process.

3.3.2. Feature Extraction Process

After several trials, the feature extraction examples are determined, and the results are shown in Figure 5.

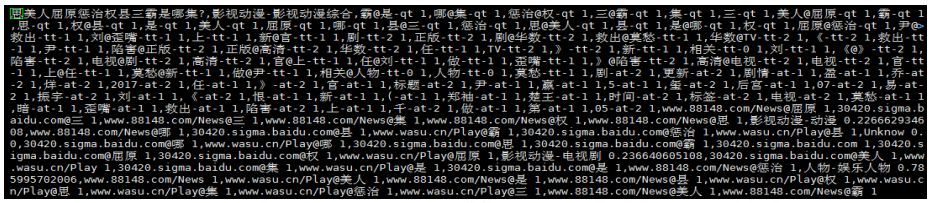

Figure 5. Sample of feature extraction

The final processing flow of the feature extraction module is as follows:

a) Feature separation

The word cell source is marked with a special symbol to separate the feature spaces of QUERY, TITLE, and ABSTRACT. As shown in Figure 5, if TERM is derived from QUERY, it is marked as "-qt". If TERM is derived from TITLE, it is marked as "-tt". If TERM is derived from ABSTRACT, it is marked as "-at". The location information is added, and the first, second, and third bars are marked with "-0", "-1", and "-2", respectively.

b) Introduction of new features

Normalize the site URL and introduce it as a new feature. If the URL level is greater than 3, keep the Level 3 and Level 2 categories; otherwise, the entire URL will be completely retained. For example, "http://money.163.com/18/0131/15/D9G4QMOC002580PL.html" is a 5-level URL directory, and the search category of clicking this URL is judged as "news". After processing, it will output " Money.163.com/18/0131-News" and "money.163.com/18-News". For different categories of the same URL, it will be aggregated, and its distribution on each category will be calculated. Finally, the two categories with the highest probability will be selected as the URL corresponding output.

c) Combination of features

Perform unigram operations on TERM in QUERY, TITLE, and ABSTRACT, perform bigram operations on TERM in

QUERY and TITLE, and perform bigram operations on normalized URL site vectors and TERM in QUERY.

d) Feature selection

For the training set, the DF algorithm is used to select features with DF greater than 6, and other features are directly filtered out.

In the feature extraction operation, the steps to greatly improve the classification effect are the feature combination and the introduction of new URL features.

4. Text Classification based on Logistic Regression Algorithm

4.1. Design and Implementation of Classification Module

4.1.1. Algorithm Selection

By observing the data to be classified daily, we find that the search traffic classification scene has large data volume, high frequency variation of the business scene, and difficulty in obtaining online feature distribution. Accuracy of the training sample cannot be guaranteed. This requires the classifier to ensure faster classification speed, fast iterative capability, and self-interpretation of model features.

The LR algorithm is primarily a two-class problem that predicts the value of a discrete dependent variable by using consistent independent variables. The LR algorithm is suitable for processing high-dimensional sparse matrices. It has the advantages of fast implementation, weighted vocabulary that can be loaded online, and unique self-interpretation of model features.

The output of the LR algorithm is a probability value of 0-1, that is, the probability that this sample belongs to a positive class. The logistic regression expression is shown in Equation (4).

$$h_\theta(x) = \frac{1}{1+e^{-z}} \tag{4}$$

Where $z = \theta^{\mathrm{T}} X$, each sample is independent of each other, the joint distribution is the product of each edge distribution, and a likelihood function is obtained. The formula is shown in Equation (5).

$$L(w) = \prod_{i=1}^{m} (\pi(x_i))^{y_i} (1-\pi(x_i))^{1-y_i} \tag{5}$$

The loss function of logistic regression is shown in Equation (6).

$$J(\theta) = \frac{1}{m} \sum_{j=1}^{m} -y_j \log\left(h_\theta(x_j)\right) - (1-y_j)\log(1-h_\theta(x_j)) \tag{6}$$

Calculate its partial derivative and obtain the logistic regression gradient drop formula as shown in Equation (7).

$$\frac{\partial}{\partial \theta_j} J(\theta) = \frac{1}{m} \sum_{j=1}^{m} (h_\theta(x_j) - y) x_j \tag{7}$$

The global optimal solution can be obtained by gradient descent.

In the business scenario, to achieve the purpose of multi-classification of text, we can use LR Softmax to directly multi-classify.

Softmax is a generalization of logistic regression on multi-classification problems [15]. In the actual development of classifiers, certain strategies are used to assist the decision classifier, and this choice depends on whether the categories are

mutually exclusive. If the categories are independent, the Softmax classifier should be used; otherwise, multiple two-category logistic regression classifiers can be used.

The formula of the Softmax classifier is shown in Equations (8) and (9).

$$\Pr(Yi = j) = \frac{e^{\beta_j X_j}}{\sum_k e^{\beta_k X_i}} \tag{8}$$

$$\sigma(z)_j = \frac{e^{zj}}{\sum_k e^{zk}} \tag{9}$$

In this scenario, the category labels for all samples are fixed and unique, so the Softmax classifier is suitable.

4.1.2. Sample Selection

a) Training set expansion

In the early stage of training, only 80,000 manual classification results and weak classifiers with low accuracy were used. It is more difficult to obtain a better training set using only the above conditions. Therefore, a number of methods are used to weakly label unlabeled samples to achieve expansion of training samples. The main methods include the following:

Semantic similarity matching [16]: Randomly extract unclassified samples on partial lines, and perform semantic similarity analysis with samples of existing classification labels. When the similarity exceeds a certain threshold (such as 0.97), the sample is labeled with corresponding labels. This sample does not compare further semantic similarities with other data.

KNN matching: In this scenario, the to-category statement and the classified statement are de-duplicated separately. At the same time, the screening threshold M is determined. When K samples with distances less than M are found, the distance comparison is stopped.

K-means clustering expansion: This scene uses cosine similarity as the distance calculation index, randomly extracts three million unlabeled search sentences for K-means processing, and outputs 10,000 clusters. After manually evaluating the intra-cluster aggregation, clusters with high content similarity are labeled with the same category.

b) Selection of verification set

In machine learning, the validation set is used to control the complexity of the model, so the size of the validation set must be guaranteed. In this scenario, 30,000 samples are randomly selected as the verification set.

c) Selection of test set

The test set evaluates the quality of the model through indicators such as accuracy and recall. The larger the sample size, the more it can reflect the effect of the classifier. However, for projects with a data size of one billionth, there is no need to use large-scale data as a test set. In this scenario, 5,000 search traffic data are randomly selected for professional manual labeling to ensure the correct classification rate, and this part of the data is directly used as a test set.

4.1.3. Model Training

Figure 6 shows the training data format obtained through feature engineering.

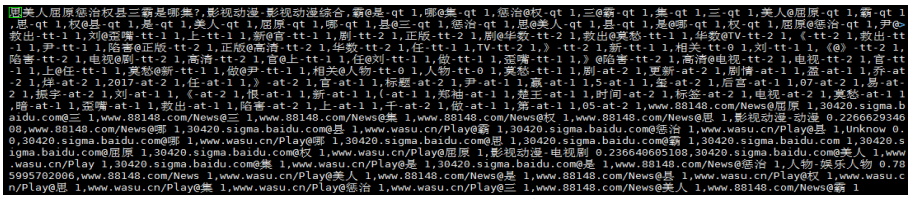

Figure 6. Training data format

Each part of the data is divided by a "," sign. The first field represents the search statement, and the second field represents the category of the manual annotation. Spaces connect features to weights.

4.2. Optimization

The most important goal of model tuning is to improve the actual benefits by changing the model. For classifiers, the most important indicators for judging the results of classification results are accuracy, recall, and recognition, which are shown in Equations (10) to (12).

$$Accuracy = \frac{The\ correct\ number\ of\ machines}{Total\ number\ of\ machines} \tag{10}$$

$$Recall\ rate = \frac{The\ correct\ number\ of\ machines}{Manually\ determine\ the\ total\ number} \tag{11}$$

$$Recognition\ rate = \frac{Total\ number\ of\ machines}{Total\ number\ of\ search\ traffic} \tag{12}$$

In Equation (10), the correct number of machine classifications refers to the total number that can be correctly classified by the traffic classification system. In Equation (11), the total number of manually categorizable can be defined as the total number of data that can be classified.

4.2.1. Category Optimization

This classifier initially has 125 category labels. The easiest way to optimize is to sort the category priorities and adjust and optimize them one by one for each category. In the optimization process, we must ensure that the current optimized category is the optimal optimization value. PV (personal visit) is the basic scale for evaluating the performance of the website. It represents the daily search traffic. The larger the PV, the more traffic is indicated for this purpose. If we optimize for such purposes, the impact on the overall results will be significant.

4.2.2. Training Set Optimization

In this model, the initial training set samples have the characteristics of low accuracy and online sample features, which can be used for error label correction, biased sample tuning, and improved classifier performance.

a) Error tag optimization

Error label optimization is mainly for label correction, purification, and other processing of training samples. In this process, the most important step is the positioning of the error label: the training set is re-classified by the first version of the trained classifier, the training set samples are retrained using different trainers, and the inconsistent classification results are manually checked.

After locating the error label, we can use the crawler modification method to crawl the name of the work in the video resource network, label it, and then modify the corresponding sample in the training set.

b) Sample distribution optimization

The size of the current training sample is far from the sample size to be classified, and it is prone to problems such as category offset. Upsampling and downsampling can be used to add and delete category samples. Upsampling is the process of adding samples to a category. For example, a copy of an existing sample is added. Downsampling involves reducing the proportion of such items by appropriately reducing the number of category samples. For example, randomly delete samples of this type.

5. Experiment and Analysis

5.1. Function Test

In practical applications, the user's search data is stored in the structured and semi-structured form on the HDFS. First, the system performs a unified format conversion to facilitate subsequent classification. This step plays a decisive role in the

quality of the classification results, so the data cleaning module and the classification module are mainly tested.

5.1.1. Classify Data on the Line

In this process, we must ensure that the test data can reflect the proportional relationship in the real environment. The purpose of this step is to verify that a good judgment can be achieved for normal data.

5.1.2. Apply Branches of the Classification System

For example, direct classification, pattern matching, and model classification are utilized, so that the objects of interest are different kinds of data. Track the classification process in all directions to test whether different use cases can correspond to the output classification results.

5.2. Long and Short Text Classification Comparison

Short text classification means that category determination is performed only according to the search sentence itself, while long text classification means that the search sentence should cooperate with the title, abstract, and URL to assist the category determination.

5.2.1. Data Preparation

Extract 5,000 data from December 20, 2017 as a test sample. The code is as follows:

```
select qid,query,display_info,clk_info
from wise_doc
where partition_dt = '20171220'
distributed by rand()
sort by rand()
limit 5000
```

5.2.2. Evaluation

The training samples are the three million samples that have been expanded as described above, and they are trained for long text classification and short text classification. After the training, the 5,000 random data obtained in the previous step are used to enter the long text classification system and the short text classification system for category judgment, and the performance of the two classification systems is evaluated. The evaluation index is shown in Equations (10) to (12).

Evaluation results: Among the 5,000 randomly selected data, a total of 4,991 can be artificially perceived, and the long text classification and short text classification results are shown in Tables 5 and 6.

Table 5. Short text classification result

	Correct number of machines	Total number of machines	Accuracy	Recall rate	Recognition rate
Primary classification	3782	4986	75.85%	76.34%	100%
Secondary classification	3325	4986	66.69%	67.12%	100%

Table 6. Long text classification result

	Correct number of machines	Total number of machines	Accuracy	Recall rate	Recognition rate
Primary classification	4140	4986	83.03%	83.57%	100%
Secondary classification	3693	4986	74.07%	74.73%	100%

As can be seen from the two tables, regardless of whether it is long text or short text classification, the accuracy and recall rate of this classification model are above 60%, and the recognition rate is 100%.

For long text classification, the accuracy rate and recall rate are 8 percentage points higher than those of short text classification, which is enough to show that the introduction of summaries, URL, and other auxiliary information as a semaphore can improve the accuracy of the classification system.

6. Conclusions

In this paper, an improved text categorization method is proposed and optimized. The experimental results show that this method is effective. It can solve the problem of low quality of training samples and improve the classification precision effectively. In the future, we will focus on solving nonlinear problems in the categorization.

Acknowledgments

This work is partIy supported by the National Key R&D Program of China (No. 2018YFB0803700) and Fundamental Research Funds for the Central Universities.

References

1. Y. W. Chen, H. Z. Wang, and H. B. Li, "A Method for Extracting Web Text Semantic Subjects based on Baidu Encyclopedia and Text Classification," *Mini-Micro Systems*, Vol. 33, No. 12, pp. 2605-2610, 2012
2. F. Sebastiani, "Machine Learning in Automated Text Categorization," *Computing Surveys*, Vol. 34, No. 1, pp. 1-47, 2002
3. M. Lalmas, "Combining Document Representations," *International Journal of Cooperative Information Systems*, Vol. 9, No. 4, pp. 427-447, 2000
4. S. Z. Wu and Z. X. Zhang, "Research on Text Representation Model based on Language Network," in *Proceedings of the 18th National Medical Information Conference of Chinese Medical Association*, 2012
5. S. R. Safavian and D. Landgrebe, "A Survey of Decision Tree Classifier Methodology," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. 21, No. 3, pp. 660-674, 1991
6. V. Bijalwan, V. Kumar, and P. Kumari, "KNN based Machine Learning Approach for Text and Document Mining," *International Journal of Database Theory and Application*, Vol. 7, No. 1, pp. 61-70, 2014
7. W. T. Zhao, L. J. Meng, and H. H. Zhao, "Application of Distributed Naive Bayesian Algorithm in Text Classification," *Measurement and Control Technology*, Vol. 6, 2016
8. W. Y. Peng, "Research on Optimization and Implementation of BP Neural Network Algorithm," in *Proceedings of International Conference on Intelligent Computation Technology and Automation*, pp. 104-107, 2014
9. O. Chapelle, V. Sindhwani, and S. S. Keerthi, "Optimization Techniques for Semi-Supervised Support Vector Machines," *Journal of Machine Learning Research*, Vol. 9, No. 1, pp. 203-233, 2008
10. F. J. Bu and X. Z. Qian, "KNN Text Classification Algorithm based on Vector Projection," *Computer Engineering and Design*, Vol. 30, No. 21, pp. 4939-4941, 2009
11. L. Zheng and D. H. Xu, "Study on Text Classification based on Improved TFIDF Algorithm," *Computer and Modernization*, Vol. 9, pp. 6-9+14, 2014
12. X. G. Pu, "A Review of Research on Automatic Text Classification Methods," *Information Science*, Vol. 26, No. 3, pp. 469-475, 2008
13. K. F. Deng, "Research on Mutual Information Feature Selection Method in Chinese Text Classification," Southwest University, 2011
14. J. F. Zhu, M. Z. Liu, and C. Q. Zhao, "Application of Regularization in Logistic Regression and Neural Networks," *Information Technology*, Vol. 7, pp. 1-5, 2016
15. B. Han, Y. J. Liu, W. X. Che, et al., "Research on Incremental Training Method of Chinese Word Segmentation based on Perceptron," *Chinese Journal of Information Science*, Vol. 29, No. 5, pp. 49-55, 2015
16. S. Namburi, "Logistic Regression with Conjugate Gradient Descent for Document Classification," Kansas State University, 2016