# Classification Method of Teaching Resources Based on Improved KNN Algorithm

https://doi.org/10.3991/ijet.v14.i04.10131

Yingbo An(✉), Meiling Xu, Chen Shen
Hebei Finance University, Hebei Baoding, China
rmgyrrezptpv7@163.com

**Abstract**—In order to effectively utilize the network teaching resources, a teaching resource classification method based on the improved KNN (K-Nearest Neighbor) algorithm was proposed. Taking the text class primary and secondary school teaching resources as the research object, combined with the domain characteristics, the KNN algorithm was improved. By measuring the sample space density, the text of the high-density area was found. Different clipping methods were proposed for both intra-class and inter-class regions. The problem of cropping in the space of multiple class boundaries was considered. Results showed that the method ensured uniform distribution of samples and reduced the time of classification. Therefore, under the Weka platform, the improved KNN algorithm is effective.

**Keywords**—Text classification, KNN, primary and secondary school teaching resources, sample cutting.

## 1 Introduction

The development of Internet technology not only allows learners to receive the highest quality education anytime and anywhere, but also allows the dissemination of knowledge to be no longer limited to books. With the popularization and promotion of digital education, the construction of online education resources in China has become increasingly mature. Various types of educational resources are abundant and large. At the same time, network resources are also facing enormous challenges. Massive educational resources are still growing at geometric multiples, and the types are complex. There is no effective organization and management. Among the various types of resources, including video, audio, pictures, and text, the number of text-based resources is the largest. In this case, how to effectively classify the teaching resources in the network is an important problem that needs to be solved urgently. In the past, manual classification was usually used to allow professional personnel to complete the classification work. In the case of a small amount of resources, this classification results are very accurate. However, with the continuous increase of the number of resources, the problem of low manual classification efficiency and the decrease of classification accuracy with the increase of working hours has become increasingly prominent.

As a kind of machine learning technology, automatic text categorization can effectively organize and manage text information. Based on a given classification model, the text to be classified is subject to certain rules. Its degree of association with each category is calculated and automatically divided into corresponding categories. This technology has a wide range of applications in information retrieval, mail filtering, and digital libraries. Text automatic classification technology saves labor costs, classification is fast, and accuracy is high. Therefore, it is regarded as the main means of classifying teaching resources.

## 2 State Of The Art

In the late 1950s, Luhn H P, an expert in text mining, first proposed the concept of word frequency statistics. Chen et al. [1] considered this to be an epoch-making study in the field. Subsequently, Samanthula et al. [2] published the first paper on text categorization and proposed the "Bayesian hypothesis", which greatly promoted the progress of text classification related research. Li et al. [3] studied the performance modeling of manufacturing personnel based on the improved KNN (K-Nearest Neighbor) algorithm. Since then, many experts and scholars have achieved certain research results in this field of technology, such as the famous intelligence scientists Spark and Salton. Since the 1980s, traditional knowledge engineering techniques have been applied in this field. According to the knowledge provided by the experts, rules were formed and the classifier was manually established. This was a good classification in some corpora. However, in the face of large-scale data sets, the method was limited. After entering the 1990s, with the rise of machine learning technology, Hong et al. [4] began to try to apply it to text classification. This classification method automatically classifies the classified text by learning on the pre-classified text set and obtaining the classification rules. This method does not require expert participation. It has higher accuracy and shorter classification time. Meng et al. [5] described and designed every detail of the text categorization implementation method, and conducted relevant experiments on the data set Reuters22173 for testing. This article was later seen as a classic in the development of text categorization. Li et al. [6] proposed a support vector machine method (SVM) based on statistical theory. The basic idea was to find the optimal high-dimensional classification hyperplane. The method can be learned based on small samples. At the same time, the robustness and classification effect were good, which has been widely concerned by experts. In addition, Anthimopoulos et al. [7] used the deep convolutional neural network to classify Lung Patterns for interstitial lung disease.

In summary, at present, the general classification algorithm was mainly used for the teaching resources of primary and secondary schools. The research and design were not combined with the characteristics of the field, and the classification effect needs to be further improved. According to the characteristics of primary and secondary school teaching resources in text class, the related research on classification algorithm was carried out. First, the basic concept of the KNN (K-Nearest Neighbor) algorithm was introduced. Then, the experimental environment was built and experimental

data was collected. Finally, the teaching resource classification method based on the improved KNN algorithm was verified. Results showed that the method was effective.

# 3 Methodology

## 3.1 Introduction of KNN algorithm

KNN algorithm (also known as K-nearest neighbor algorithm) is one of the most commonly used classification algorithms at present. It is ideal for solving multi-category problems. The core idea is that in the feature space, if the K samples closest to the sample to be classified mostly belong to a certain category, the sample to be classified also belongs to this category.

For example, the training space contains two categories of blue squares and red triangles, and the green text is the sample to be classified. When K=3, in the three samples closest to the green sample, there are two red and one blue, and the green sample should be divided into red triangles. When K=5, in the three samples closest to the green sample to be classified, there are three blue and two red, and the green sample is divided into blue squares. From this, it can be seen that the selection of the K value directly affects the final result of the KNN classification.

Text similarity is used to measure the distance between two texts. Commonly used similarity calculation methods include the Euclidean distance method and the angle cosine method.

The Euclidean distance method is as follows:

$$D(d_i, d_j) = \sqrt{\sum_{k=1}^{n} (W_{ik} - W_{jk})^2} \tag{1}$$

In the formula, $d_i$ and $d_j$ represent any two texts i and j in the training set. n is the total dimension of the feature vector. $W_{ik}$ and $W_{jk}$ represent the corresponding feature item weights in the text vectors of the text $d_i$ and $d_j$.

The smaller the value of the Euclidean distance D, the higher the similarity between the two texts. On the contrary, the similarity is low. The Euclidean distance method is a relatively simple method of calculating similarity. The amount of calculation is small, but the results are often not good enough.

The angle cosine method is as follows:

$$Sim(d_i, d_j) = \frac{\sum_{k=1}^{n} W_{ik} \times W_{jk}}{\sqrt{(\sum_{k=1}^{n} W_{ik}^2)(\sum_{k=1}^{n} W_{jk}^2)}} \tag{2}$$

The meaning of the parameters in formula (2) is the same as in the Euclidean distance formula (1).

The larger the angle cosine value Sim ($d_i$, $d_j$), the smaller the vector angle of the two texts, which indicates that the similarity of the two vectors is high. Conversely, the smaller the angle cosine value Sim ($d_i$, $d_j$), the larger the vector angle of the two texts, the lower the similarity. In addition, the range of the cosine of the included

angle should be [0, 1]. If it is not within the range, the calculation result is incorrect. When Sim=0, the two articles are completely unrelated; when Sim=1, the two articles are identical. At present, the similarity calculation in the text classification system generally adopts the angle cosine method.

The specific steps of the KNN algorithm are introduced, as shown in Figure 1.

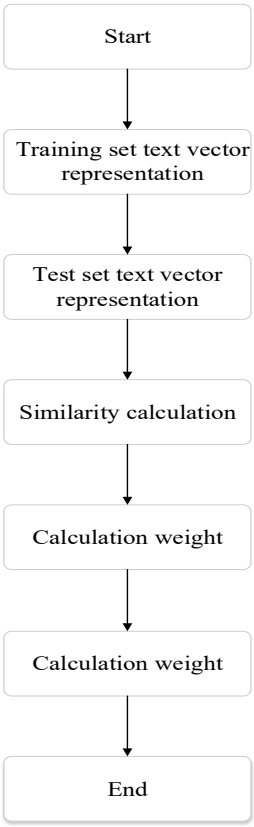


**Fig. 1.** Flow chart of KNN algorithm

**First,** the training set text is represented by a vector space model. The feature weight of each dimension is the result of the TF-IDF_ATC calculation.

**Second,** the test set text is also represented by a vector space model. The feature weight of each dimension is also the result of TF-IDF-ATC calculation.

**Third,** the spatial distance between the text to be classified in the test set and each text in the training set is calculated, that is, the text similarity. In the text similarity algorithm, the cosine similarity algorithm which is more suitable for KNN is selected, as shown in formula (3).

$$Sim(X, D_j) = \frac{\sum_{i=1}^{n} W_i \times W_{ji}}{\sqrt{(\sum_{i=1}^{n} W_i^2)(\sum_{i=1}^{n} W_{ji}^2)}} \tag{3}$$

In the formula, X is the text to be classified, and $D_j$ is a text j in the training set. $W_i$ represents the i-th feature item weight of the text vector X to be classified. $W_{ji}$ represents the i-th feature weight of a text j in the training set. n is the total number of dimensions of the text feature vector.

Fourth, the Sim(X, $D_j$) values are arranged in descending order, and K training set texts having the highest similarity to the text X to be classified are selected. At present, there is no good way to directly determine the optimal K value. The K value can only be adjusted through experiments, and the appropriate K value is determined according to the test classification effect.

Fifth, the distribution of the selected K texts in each class should be counted, and the text to be classified should be classified into the category of most texts. There are many teaching resources in primary and secondary schools. It is easy to appear that several categories contain the same number of texts. Therefore, according to the category of the K most recent texts, the weight of the text to be classified belongs to each category, as shown in formula (4).

$$P(X, C_i) = \sum Sim(X, D_j) P(D_j, C_i) \tag{4}$$

Sim(X, $D_j$) is the similarity between the text to be classified and the text of the training set. P($D_j$, $C_i$) is a category attribute function. When the text X belongs to the class Ci, P($D_j$, $C_i$) = 1; otherwise, P($D_j$, $C_i$) = 0. Finally, the text X to be classified is divided into categories with the largest P(X, $C_i$) value.

### 3.2 KNN algorithm analysis

According to the characteristics of the teaching resources in primary and secondary schools, two shortcomings of the traditional KNN algorithm are proposed:

**First**, when the number of training set texts is large, the computational overhead of the KNN algorithm is large.

**Second**, the sample distribution between the classes of primary school teaching resources in the text category is severely uneven. There is a misclassification phenomenon in the KNN algorithm classification.

Taking junior high school as an example, the distribution of resources in the resource pool of an ideal cloud platform is shown in Table 1.

**Table 1.** Distribution of platform resources

| Subject name | Number of resources |
|---|---|
| Mathematics | 1081 |
| Physical | 872 |
| Chemistry | 1408 |
| Language and literature | 5908 |
| History | 5266 |
| Geography | 1152 |

From the table, it can be clearly seen that the number of liberal arts resources is far greater than the number of science resources, especially between subjects with large

differences, such as mathematics and language, which can even reach 5-6 times. During the classification process, the samples between the various text classes of the training set are unevenly distributed. The effectiveness and efficiency of the KNN classification algorithm is greatly reduced. It can be clearly seen that the text to be classified should be classified into category 2. However, when k=10, since the density of category 1 is much larger than category 2, when using the KNN classification algorithm, 6 out of the 10 most recent samples selected belong to category 1, and 4 texts belong to category 2. The text to be classified is classified into category 1 to produce an incorrect classification result.

In the next section, in response to the above shortcomings, the corresponding improvement strategy for KNN algorithm is proposed, and the specific improvement strategy is deeply studied.

### 3.3 Design of KNN improved algorithm based on density cutting scheme

At present, there are two main ways to reduce the computational complexity of the KNN classification method: one is to reduce the time to find the nearest neighbor of the sample to be classified by optimizing the retrieval algorithm. The other is to select some representative samples in the original training sample set as new training samples, or delete some samples in the sample set. Then, the remaining samples are taken as new training samples, and the training sample set is reduced. As a result, the calculation work is reduced.

This study chose the second method. An improved KNN algorithm for applying density tailoring scheme is proposed. The time complexity of the KNN algorithm is reduced. At the same time, the problem of the classification accuracy rate due to the uneven distribution of sample density is solved in the training concentration.

The following basic concepts are defined, which facilitate the measurement of the density distribution of the training sample space. Finally, the clipping of the sample space is implemented.

A training sample set S is given. The definition is as follows:

Assuming that X and Y are two samples in the sample set S, Dist (X, Y) is used to represent the distance between X and Y. For any X∈S, its ε neighborhood is as shown in equation (5).

$$N_{\varepsilon} = \{Y \mid Dist(X,Y) \le \varepsilon, Y \in D\} \tag{5}$$

The circle X with the text X as the center and Dist (X, Y) as the radius is the ε neighborhood of X. Supposing that the ε neighborhood of X contains the number of samples of Nεx, and the ε neighborhood of X contains the minimum number of samples of MinPts, then:

When Nεx=Minpts, the ε neighborhood of X is a uniform density region;
When Nεx>Minpts, the ε neighborhood of X is a high-density region;
When Nεx < Minpts, the ε neighborhood of X is called a high density region.

Supposing that the sample set S contains {C1, C2, C3, Cn} for a total of n sample categories, and the classes and classes do not intersect each other. For any X∈Ci, if all text in the ε neighborhood of X belongs to the Ci class, X is in the intra-class region. Otherwise, if there are other categories of samples in the ε neighborhood of X in addition to the Ci-type text, X is in the junction area.

The high density area is divided into two cases. Among them, the red circle represents the inner region, and the blue circle represents the boundary region. The high-density cropping method for both cases is illustrated:

**First, intra class region clipping (red):** First, supposing that Y is in a high-density text area, and any other text in the ε neighborhood of Y is sequentially determined, such as whether the ε neighborhood of Z is at a high density. If it is at a high density, the text is cropped and N-- until the Y neighborhood N = Minpts, thereby making the Y neighborhood density uniform.

As can be seen from the above method, for the text Y in the high-density region, the text in the high-density region in the ε neighborhood of Y is removed as much as possible. This not only makes the density of Y relatively uniform, but also makes the text in the vicinity thereof more uniform. Minpts=3, Z is a text with a high density in the ε neighborhood of Y, so Z will be cut off. Similarly, there will be 5 texts in the ε neighborhood of Y that will be cropped, and N=3.

**Second, border area clipping (blue):** Assuming that the total number of categories of the ε neighborhood of X is C, an integer Hpts is selected as the minimum number of samples of the boundary region in place of the previously defined Minpts. Compared to the inner class region samples, the samples in the boundary region have a greater contribution to the classification. Therefore, for selected Hpts, Htps ≥ Minpts > 0.

If C>Hpts, one sample is reserved for each class (C1, C2, Ci).

If C<Hpts, the number of samples retained by each class in the ε neighborhood is determined, and the calculation method is as follows:

The number of texts Tn for each sample class in the ε neighborhood is sorted in descending order;

The parameter RA=Hpts/Ciε (surplus) is set;

The text class of Tn ranked in the top RA is determined, and the sample number REci=(Hpts/C)+1 is reserved;

The number of reserved samples for the remaining text classes is: REci=Hpts/C.

Whether the other arbitrary text is in the high-density region is determined sequentially in the ε neighborhood of the sample X. If it is in a high-density region, the sample is cropped until the number of samples Tn is the same as the number of samples required to be retained in the previous step, and N = Htps.

The main process of the density cutting scheme is shown in Figure 2.

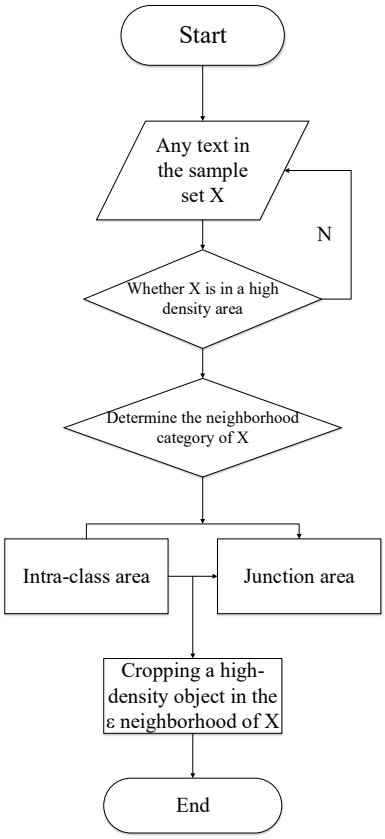


**Fig. 2.** Flow chart of density cutting

Input: training sample set S, neighborhood radius ε, integers greater than 0 MinPts and HPts.

Output: Cropped training sample set P.

Steps:

P={};

FOR EACH X∈S DO

BEGIN

IF X is in the inner region THEN

IF Nεx≤Minpts THEN

P=P+{X};

ELSE

IF Nεx≤Hpts THEN

P=P+{X};

END;// Keep training samples in low density areas

FOR EACH(X∈S&X∈P)DO

BEGIN

R(X)={Y|Y∈Nεx, Y∈P};// Sample within the ε neighborhood of X
H(X)={Y|Y∈R(X), Y is a high density area text};
IF X is in the inner region THEN
WHILE Nεx>Minpts DO
BEGIN
L=arg max Nεti, ti∈H(X);
Y∈TL, TL∈H(X);
IF NεY>Minpts THEN
S=S-{Y};
ELSE
Break;
END;// High density cutting in the inner region
ELSE IF X is in the boundary area
IF Cεi>Hpts// The total number of categories in the ε neighborhood of X is Cεi
WHILE Keep at least one text DO per class
BEGIN
L=arg max Nεti, ti∈H(X);
Y∈TL, TL∈H(X);
S=S-{Y};// Keep at least one sample END per class;
ELSE IF Cεi <Hpts
RA=Hpts% Cεi;
WHILE Nεx>Hpts DO
BEGIN
IF T, Ranked in the top RA
THEN
REεi=Hpts/ Cεi +1;
ELSE
RE εi =Hpts/ Cεi;
L=arg max N, ti∈H(X);
Y∈TL, TL∈H(X);
IF N CεY>REεiTHEN
S=S-{Y};
END;// High density cutting in the boundary area
P=P+N;
END;

As can be seen from the above description, the time complexity of the sample cutting process is O (2n). The time complexity of the traditional KNN algorithm is O (n2), and the time complexity of sample clipping is non-exponential. The time complexity of the KNN classification algorithm is negligible, and the time complexity of the whole algorithm is still O (n2). At the same time, n becomes smaller as the cutting process of the training sample set becomes smaller. Therefore, the time complexity of the improved algorithm becomes smaller.

In addition, the three main parameters in the density cropping scheme need to be determined, namely Hpts, Minpts and ε. According to the experimental results, it can

be seen that when the value of Hpts is taken as 5% to 8% of the average number of samples in the category, a relatively good effect can be obtained. The range of values of Minpts is an integer greater than 0 and less than or equal to Hpts. The best Minpts value is obtained through experimental debugging. The value of ε is calculated according to formula (6).

$$\varepsilon = Density_{Hpts}(S) = \frac{1}{D}\sum_{i}^{|D|} Dist_{Hpts}(X_i) \tag{6}$$

Among them, DensityHpts(S) is the average neighborhood radius of the training set S with a minimum number of samples of Hpts. Distk(X) represents the distance from the kth nearest neighbor to X of the sample X in the sample set S.

## 4 Results Analysis and Discussion

Weka is an open source project under the Java platform at the University of Waikato, New Zealand. It has the characteristics of cross-platform, support structure text, and database interface. In addition, many of today's most advanced machine learning and data mining algorithms are combined. It can effectively complete tasks such as preprocessing, classification, clustering, correlation, and visualization. Since Weka is an open source project, it is very convenient to perform secondary development of user algorithm embedding and parameter modification on the basis of classical algorithms. According to the previous research results, using the open source provided by Weka, the TF-IDF weight calculation method and KNN algorithm are rewritten to implement the improved algorithm.

### 4.1 Construction of the experimental environment

- Hardware environment: Windows10 64-bit + Intel Core i7 4720HQ+8GB
- Software environment: Weka3.8.2+Eclipse+sqlserver2010.

When downloading the Weka installer from the official website, there are two versions of weka-3-8-2-x64.exe and weka-3-8-2jre-x64.exe. Weka-3-8-2-x64.exe only installs Weka, and weka-3-8-2jre-x64.exe installs java virtual machine in addition to Weka.

Environment variables need to be configured after the installation is complete. In the system environment variable, the CLASSPATH is found to add the path to weka. For example, if Weka is installed on the D drive, D:\weka-3-8\weka.jar is added. Users need to download Eclipse and sqlserver2010 themselves. The main interface of Weka is shown in Figure 3.


**Fig. 3.** The main interface of Weka

The steps to import Weka source code in Eclipse are as follows:

- **First,** Weka was downloaded. The file weka-src.jar is decompressed, including three folders lib, src and META-INF and two other files;
- **Second**, under Eclipse, the java project is created, which is named weka, and a new package named weka is created under src;
- **Third,** in this project, import-->File System-->select.../weka/src/main/java/weka, and import all;
- **Fourth,** project import library file, biuldpath-->addexternaljar-->select lib java-cup.jarJFlex.nit.jar;
- **Fifth,** weka.gui.main was successfully run.
- In addition, since Weka does not support Chinese by default, the configuration file RunWeka.ini needs to be changed. This configuration file is in the directory after Weka is installed. This file is opened to find the fileEncoding=Cp1252 line to change to fileEncoding=utf-8, as shown in Figure 4.

```
1 # The file encoding; use "utf-8" instead of "Cp1252" to display UTF-8 characters in the
2 # GUI, e.g., the Explorer
3 fileEncoding=utf-8
```

**Fig. 4.** Modification of configuration file RunWeka.ini

Since Weka itself comes with English word segmentation and no Chinese word segmentation function, Chinese word segmentation operations need to be performed. In Eclipse, the mature word breaker tool is called. After getting the result of the word segmentation, it is called directly by Weka.

The Chinese Academy of Sciences' NLPIR system (formerly ICTCLAS) was selected. It is very simple and convenient.

First, a JAVA project was created. The jna jar package (which can be copied from the sample\JnaTest NLPIR\lib folder in the download package) is imported. The Data

folder in the download package is copied to the project root directory. Then, the NLPIR.dll and NLPIR.lib files in the folder corresponding to the operating system in the lib folder are copied to the newly created source folder in the system root directory, as shown in Figure 5.

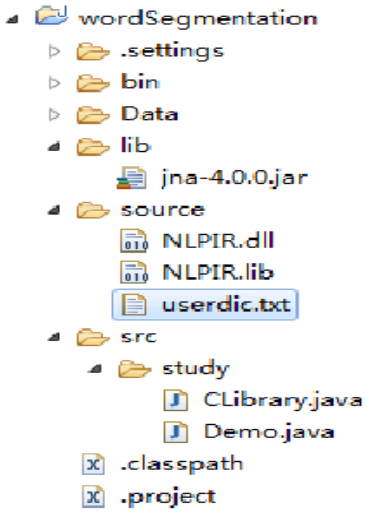


**Fig. 5.** Diagram of JAVA project for NLPIR system

An interface is created to inherit jna's Library interface. In the interface, an implementation of a series of NLPIR functions is defined. Finally, after the interface is instantiated in the class, the function can be called.

The result of the word segmentation obtained through the NLPIR system is stored in a .txt file format. However, since Weka can only accept .arff format files, the file type needs to be converted. This is achieved by learning from mature code already on the network.

### 4.2 Experimental data set

The experimental data set is a self-built corpus. The corpus contains 3,000 texts of various types and is unevenly distributed. There are 10 categories in total: mathematics, Chinese, English, physics, chemistry, biology, politics, geography, history, and others. Among them, the training set has a total of 2,100 texts and 900 test sets.

### 4.3 Testing and analyses of the algorithm

Based on the open source code provided by the Weka platform, the second development is carried out, the improved algorithm is implemented. Its performance is tested, and the classification effect before the algorithm improvement is compared and analyzed.

The purpose of the experiment was to find the optimal K value.

The corresponding class name of the KNN algorithm in Weka is IBk. Through the GUI, the Classify is selected and click the choose button to find IBk under weka/classifiers/lazy. Then, the red box is clicked to set the K value.

Experiment 1: The selected feature dimension was 500, and K = {5, 10, 15, 20, 25, 30, 35} were respectively tested. The experimental results are shown in Table 2.

**Table 2.** The effect of the K value on the macro F1 value

| K | 5 | 10 | 15 | 20 | 25 | 30 | 35 |
|---|---|---|---|---|---|---|---|
| Macro F1 value | 75.83 | 77.14 | 78.89 | 77.62 | 77.11 | 76.64 | 76.57 |

Analysis of results: It can be seen from the experimental data that when K takes 5~15, the macro F1 value shows an upward trend. When K=15, the macro F1 gets the maximum value. When K>15, the K value gradually decreased. Therefore, based on the experimental data set of this paper, when K=15, the classification effect is the best. At present, there is no particularly good way to determine the value of K. The main reason is to determine the K value that is most suitable for the selected data set based on experience and continuous experimentation.

Analysis of TF-IDF and improved algorithm TF-IDF_ATC results

The purpose of the experiment is to compare the classification effects before and after the improvement of TF-IDF.

Experimental environment: 2100 training texts, 900 test texts, feature dimensions of 500, and K values of 15.

Experiment 2: The TF-IDF method is defined in Weka's StringToWorldVector class, which is located in weka->filters->unsupervised->attribute->StringToWordVector.

The comparison of the experimental results before and after the improvement is shown in Table 3.

**Table 3.** Comparison of single class classification results

| Category | TF-IDF | | | TF-IDF_ATC | | |
|---|---|---|---|---|---|---|
| | ***Recall rate*** | ***Accuracy*** | ***F1 value*** | ***Recall rate*** | ***Accuracy*** | ***F1 value*** |
| Mathematics | 78 | 83.71 | 80.75 | 80 | 85.26 | 82.55 |
| Language and literature | 82 | 79.16 | 80.55 | 89 | 84.35 | 86.61 |
| English | 78 | 96.50 | 86.27 | 78 | 97.65 | 86.73 |
| Physical | 74 | 87.14 | 80.03 | 76 | 88.34 | 81.71 |
| Chemistry | 75 | 85.67 | 78.83 | 78 | 86.98 | 82.25 |
| Biological | 72 | 82.55 | 76.91 | 74 | 85.32 | 79.26 |
| Political | 76 | 80.34 | 78.11 | 74 | 83.42 | 78.43 |
| Geography | 70 | 84.52 | 76.58 | 72 | 84.78 | 77.87 |
| History | 82 | 78.69 | 80.31 | 80 | 82.67 | 81.31 |
| Others | 68 | 73.28 | 70.54 | 72 | 74.15 | 73.06 |

In theory, English subjects and other subject texts are easy to distinguish, and the classification accuracy rate can reach 100%. However, in the experiment, the accuracy rate is low. This is because some English subjects contain only a small amount of

English, and most of them are Chinese content. For example, in a lesson plan that explains words, an English word needs to have a paragraph of Chinese to explain its meaning. The overall experimental results are shown in Table 4.

**Table 4.** Comparison of macro F1 values

| Weight calculation method | Macro F1 value |
|---|---|
| TF-IDF | 78.89 |
| TF-IDF_ATC | 80.98 |

From the experimental data, it can be clearly seen that the TF-IDF_ATC algorithm has a slight decrease in the recall rate in politics and history. In addition to the English recall rate, the accuracy and recall rate of other categories have increased to some extent. Moreover, the F1 value as a comprehensive evaluation indicator has also been significantly improved in each category. On the whole, the macro F1 value is also increased by about 2 percentage points compared with the TF-IDF algorithm before the improvement. It shows that the improved weight calculation method TF-IDF_ATC has better weight distribution ability than traditional TF-IDF, and has better classification effect.

Analysis of experimental results before and after KNN algorithm improvement

The purpose of the experiment is to compare the classification effect and time before and after the sample space is cropped, and determine the parameters Minpts.

Experimental environment: 2100 pieces of training text in the corpus are used as training set S, 900 pieces are used as test text, feature dimension is 500, and K value is 15. Hpts takes 5% to 8% of the average of the category samples, and Hpts=11. For the training set S, ε=DensityHpts(S) is taken, and the Minpts take values from 1 to 11. The training sample set is cropped using a sample cropping algorithm. The results of the cropping are shown in Table 5. Among them, the crop ratio = the number of cropped samples / the total number of training set samples.

**Table 5.** Cropping of training set S

| Minpts | Number of crops | Crop ratio (%) |
|---|---|---|
| 1 | 1037 | 49.4 |
| 2 | 874 | 41.6 |
| 3 | 743 | 35.4 |
| 4 | 664 | 31.6 |
| 5 | 611 | 29.1 |
| 6 | 562 | 26.8 |
| 7 | 518 | 24.7 |
| 8 | 481 | 22.9 |
| 9 | 449 | 21.4 |
| 10 | 423 | 20.1 |
| 11 | 405 | 19.2 |
| **Total number of samples: 2100** | | |

As can be seen from Table 5, the proportion of cropping decreases as the Minpts increases. Even when Minpts=Hpts, the sample space crop ratio can still reach about 20%. The amount of calculation of the KNN classification is effectively reduced, and the classification time is shortened. This is not a sacrifice of the classification effect.

The classification macro F1 value comparison is performed using the training set before and after the cropping, respectively. When Minpts takes 8~11, the KNN improved algorithm of density cropping scheme can be used to obtain better macro F1 value. A better classification effect is obtained than before the improvement. Analysis of experimental results: Considering the two factors of classification time and classification effect, it is found that when the value of Mintps is in the range of 8 to 11, the performance of the improved algorithm is the best. At this time, the improved KNN algorithm applying the density cropping scheme not only reduces the classification time, but also improves the classification effect to some extent, which proves the effectiveness of the improved algorithm.

## 5 Conclusion

The key theories and techniques involved in the text categorization process were studied. At the same time, the steps and basic flow of text categorization are introduced, including text preprocessing, feature extraction, weight calculation and so on. This laid a solid foundation for the follow-up work. The text preprocessing process is improved by combining the characteristics of text-based primary and secondary school teaching resources. Resource characteristics were analyzed. The corpus is built through category partitioning, resource filtering, and uniform text formatting. The text preprocessing process was improved. An improved strategy is proposed for the KNN algorithm. Aiming at the problems of traditional KNN algorithm in the classification of teaching resources in primary and middle schools, an improved KNN algorithm based on density tailoring scheme is proposed. Samples of high-density regions in the sample space are cropped before the KNN algorithm is executed. The problem of misclassification caused by uneven distribution of sample spatial density is solved. At the same time, the time complexity of the KNN classification is reduced. The appropriate parameters K and Minpts were determined by comparison experiments, and the effectiveness of the improved algorithm was verified. The results show that the classification method of primary and secondary school teaching resources based on KNN algorithm is feasible and effective.

## 6 Acknowledgement

This research is supported by the fund project of application of Intelligent Financial Technology Research and Development Centers of Hebei.

## 7 References

[1] Chen R, Chen F, Sun Y. Research on Automatic Text Classification Algorithm Based on ITF-IDF and KNN. Applied Mechanics & Materials, 2015, vol. 713-715, pp. 1830-1834 https://doi.org/10.4028/www.scientific.net/AMM.713-715.1830

[2] Samanthula B K, Elmehdwi Y, Jiang W. k-Nearest Neighbor Classification over Semantically Secure Encrypted Relational Data. IEEE Transactions on Knowledge & Data Engineering, 2015, vol. 27(5), pp. 1261-1273. https://doi.org/10.1109/TKDE.2014.2364027

[3] Li N, Kong H, Ma Y, et al. Human performance modeling for manufacturing based on an improved KNN algorithm. International Journal of Advanced Manufacturing Technology, 2016, vol. 84(1-4), pp. 1-11. https://doi.org/10.1007/s00170-016-8418-6

[4] Hong Z, Li Q, Yong Q. Loyalty prediction method based on improved nearest neighbor algorithm. Journal of Nanjing University of Science & Technology, 2017, vol. 41(4), pp. 448-453

[5] Meng Meng, Chun-fu Shao, Yiik-diew Wong, et al. A two-stage short-term traffic flow prediction method based on AVL and AKNN techniques. Journal of Central South University, 2015, vol. 22(2), pp. 779-786. https://doi.org/10.1007/s11771-015-2582-y

[6] Li, Q, Zhang, Z, Lu, W, et al. From pixels to patches: a cloud classification method based on bag of micro-structures. Atmospheric Measurement Techniques, 2016, vol. 8(10), pp. 10213-10247. https://doi.org/10.5194/amtd-8-10213-2015

[7] Anthimopoulos M, Christodoulidis S, Ebner L, et al. Lung Pattern Classification for Interstitial Lung Diseases Using a Deep Convolutional Neural Network. IEEE Transactions on Medical Imaging, 2016, vol. 35(5), pp. 1207-1216. https://doi.org/10.1109/TMI.2016.2535865

## 8 Authors

**Yingbo** An (corresponding author) is a lecturer Hebei Universities, Hebei Finance University (rmgyrrezptpv7@163.com)

**Meiling Xu** is a lecturer Hebei Universities, Hebei Finance University (Meiling Xu@163.com)

**Chen Shen** is a lecturer Hebei Universities, Hebei Finance University (Chen Shen @163.com)

Article submitted 13 November 2018. Resubmitted 03 January 2019. Final acceptance 18 January 2019. Final version published as submitted by the authors.