Research and Application of Webpage Information Recognition Method Based on KNN Algorithm

Ziyun Deng

***Abstract*—To identify informational webpages among massive webpages crawled, we used the k-nearest neighbor (KNN) algorithm. The dichotomous classification of webpages using the KNN model was divided into four steps: the extraction and annotation of feature data, the normalization of feature data, the training and evaluation of the KNN model, and the prediction of webpage types. Two key points were considered when using the KNN algorithm. One was the extraction of appropriate feature data items from the webpages. The other was the determination of the value of k with maximum accuracy. We proposed the use of four feature data items to express webpage features: the number of continuous large texts, the number of pure texts, the proportion of pure texts, and the average length of pure texts. We also proposed calculations for each of these items. The experiments showed that the KNN model using the four feature data items had a good dichotomous classification ability, and the accuracy reached 97.5% of the peak when the value of k was 7**

***Index Terms*—KNN model, Information webpage, Identification method, Feature data item, Accuracy.**

I. INTRODUCTION

AFTER crawling massive webpages, we need to classify and analyze the data collected using big data technology [1]. Massive webpages are first divided into two types: informational webpages and noninformational webpages. Then, the titles and texts of the informational webpages are extracted. These texts can be used for analyzing public opinion, determining popular topics and other functional applications that use technologies including semantics and pattern matching [2]. Therefore, we need to determine an appropriate method for classifying massive webpages into informational webpages and noninformational webpages.

II. RELATED WORKS

Currently, there are many methods that can be used for dichotomous classification. These methods are divided into three types.

1. The first type involves direct classification according to the characteristics of webpages. For example, certain types of webpages contain certain strings or tags [3]. If we want to identify the informational webpages of multiple websites, it is difficult to determine common features to classify them directly [4]. Therefore, this approach is not recommended.

2. The second type involves the use of classical machine learning algorithms. Many algorithms can be used, including linear regression [5], logistic regression [6], k-nearest neighbor (KNN) [7], and back propagation (BP) neural networks [8].

The third type involves the use of deep learning algorithms. For example, we can use convolutional neural networks (CNNs) [9]. If deep learning algorithms are used, then considerable training data are needed. Due to the complexity and number of parameters in the deep learning network, the training time is usually long [10].

Regardless of whether classical machine learning or deep learning algorithms are used in identifying informational webpages [11], three main problems need to be considered to select appropriate algorithms for dichotomous classification.

1. The feature data items of webpages must be constructed. These items can be used for the dichotomous classification of informational and noninformational webpages.

2. A dataset must be labeled in advance.

3. Priority should be given to supervised algorithms. We should train the model and determine the parameters of the model in advance. The reason for this choice is that clustering algorithms cannot ensure that the two classifications are informational webpages and noninformational webpages [12].

Therefore, based on solving these three problems, we choose the relatively simple and easy-to-use classical machine learning algorithm KNN.

III. RESEARCH AND APPLICATION IDEAS

According to the discussion of related works, we use the KNN algorithm to identify informational webpages. The workflow of this model is shown in Figure 1.

The KNN algorithm is a supervised and lazy loading algorithm [13] that requires some labeled data in advance [14]. As our focus is on application innovation, we do not intend to propose a new machine learning algorithm. Instead, we use the mature KNN algorithm. The following steps are taken for the method and experiment.

1. Feature data are extracted, and the results are labeled. First, feature data items used to identify informational webpages are proposed according to the application of dichotomous classification. Next, programs for extracting the feature data are developed. In addition, the dichotomous classification results of webpages need to be labeled in advance [14]. The feature data and classification results can be used as the training and test data of the KNN model.

2. The feature data in the experiment are normalized. It is necessary to control the feature data within the range of [0,1] [15]. During normalization, it is also necessary to record the maximum values of feature data items. These maximum values are used to recover the feature data of webpages.

Manuscript received August 15, 2021; revised June 9, 2022. This work is supported by the Natural Science Foundation of Hunan Province (No. 2020JJ7091).

Ziyun Deng is a professor of the College of Hunan Business, Changsha Commerce & Tourism College, Changsha, 410116, China (phone: +86 13874921889; e-mail: dengziyun@126.com).

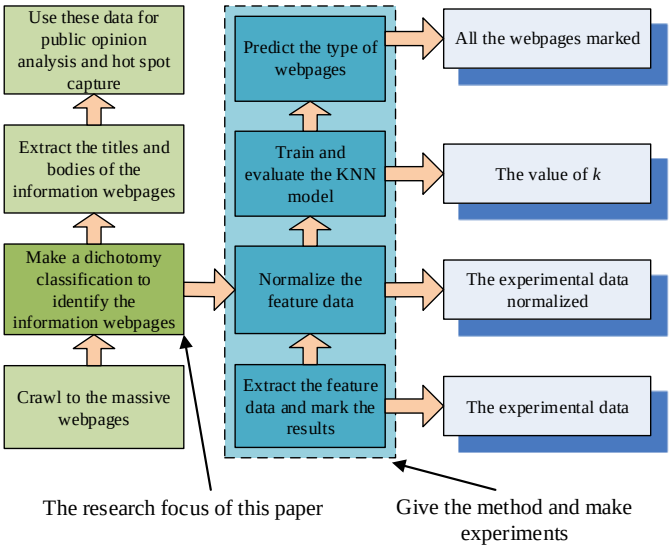

Fig. 1. The focus of this paper and work ideas

3. The KNN model is trained and evaluated. The appropriate value of k is obtained according to the maximum accuracy [16].

4. The types of webpages are predicted. According to the value of k and the experimental data, the KNN model is utilized to predict the types of webpages.

IV. KNN ALGORITHM

A. Concept of the KNN algorithm

The core concept of the KNN algorithm is simple and easy to understand. The only parameter to be determined is the value of k, which represents the k data sample points closest to the feature data points of the webpage to be predicted [17]. In general, k is set as an odd number to judge which type belongs to the dichotomous classification according to the voting rules. If the k closest training data belong to a certain type, then the webpage is predicted to belong to this type [18]. The idea is illustrated in Figure 2.

Suppose there are two webpages, as shown in Figure 2, and each webpage has two feature data items. In Figure 2(a), the value of k is three. The point of the webpage to be predicted is connected with the points of the nearest three classified webpages. Two connected points belong to informational webpages. The prediction result shows that the webpage belongs to informational webpages. Similarly, in Figure 2(b), the value of k is five. The prediction result shows that the webpage is a noninformational webpage.

B. Methods of calculating distance

In the application scenario of dichotomous classification to identify webpages, the KNN algorithm adopts the Euclidean distance as the distance measurement [19]. Accordingly, if webpages x_1 and x_2 have n feature data items, the distance is calculated by

$$distance = \sqrt{\sum_{i}^{n} (x_{1i} - x_{2i})^2} \quad (1)$$

In Formula 1, x_{1i} represents the ith feature data item of x_1. x_{2i} represents the ith feature data item of x_2.

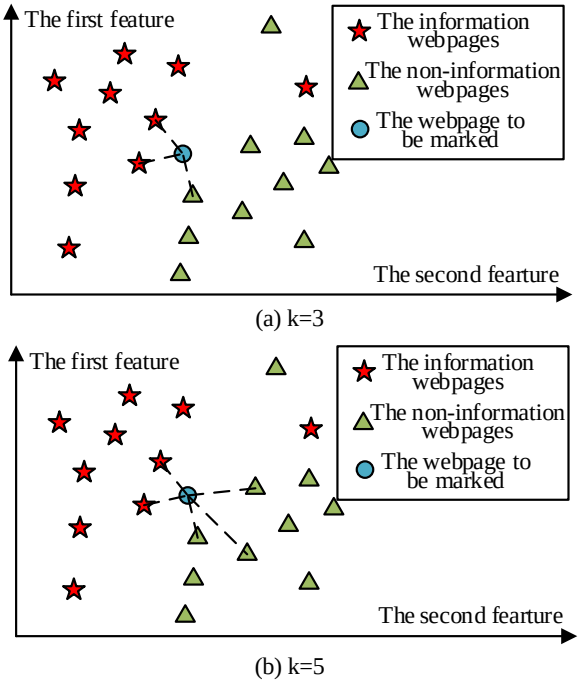

Fig. 2. Prediction of the KNN algorithm under different k values

C. Appropriate value of k

What is the appropriate value of k? We take the value with the highest accuracy. How can a more stable accuracy be determined? The K-fold cross-verification approach can be used to determine the accuracy of the algorithm. Using K-fold cross-verification, the experimental data are first divided into K parts. The parameter K in K-fold cross-verification and the parameter k in the KNN algorithm are two different parameters. Then, K-1 parts of the experimental data are used as training data, and one part of the experimental data is used as verification data [20]. The experimental data are randomly divided K times and verified K times. According to the accuracy of K verification results, we take the average accuracy as the final accuracy [20].

V. PROCESSES OF APPLYING ALGORITHM

The process of identifying informational webpages by dichotomous classification is analyzed in detail below and shown in Figure 1 as a series of steps.

A. Feature data extraction and annotation results

The labeled results indicate whether a webpage is an informational webpage or a noninformational webpage. Four feature data items, which are described below, can be used to clearly distinguish between these types of webpages.

1. Number of consecutive large texts

The term large text refers to text on a webpage with a length of more than 50 characters. The nontext of an informational webpage includes the link text, script text, select item text, form text, and input item text. By using the XPath expression "//body/text()" on the webpage, all texts of tag contents can be extracted as a list $alltextlist$ according to the tag order in the HTML document of the webpage. As shown in Table I, the nontext of the informational webpage can be extracted through different XPath expressions.

TABLE I
NONTEXT OF INFORMATION WEBPAGES

List	XPath expression	List name
The list of script text	//script/text()	$scriptTextList$
The list of form text	//form/text()	$formTextList$
The list of select item text	//option/text()	$optionTextList$
The list of input item text	//input/text()	$inputTextList$
The list of link text	//href/text()	$hrefTextList$

The list $pureAndHrefTextList$ is used to represent the possible title text and body text of the informational webpage, and it can be obtained by the following calculation.

$$\begin{aligned} pureAndHrefTextList = allTextList \\ -scriptTextList - formTextList \\ -optionTextList - inputTextList \end{aligned} \tag{2}$$

The subtraction operation in Formula 2 is the subtraction operation for some sets.

Large texts are continuous if there are no link texts between two large texts. We believe that the body content of the informational webpages should have at least two consecutive large texts. The pseudocode of the algorithm to obtain the number of consecutive large texts is shown in Algorithm I.

In Algorithm I, the text in the list $pureAndHrefTextList$ is large text if the length of the text reaches fifty characters and the text is not a link text. In the second for loop body of the list $pureAndHrefTextList$, if the current element is the first element of the list, as long as the first element and its next element are also large text, the number of consecutive large texts will be increased by one. If the current element is the last element of the list $pureAndHrefTextList$, as long as the last element and its previous element are large text, the number of consecutive large texts will be increased by one. If the current element is an intermediate element of the list $pureAndHrefTextList$, in addition to the fact that the current element is a large text, if either the previous element is a large text or the next element is a large text, then the number of consecutive large texts will be increased by one.

2. Number of pure texts

Pure text refers to the title and body content of informational webpages. We believe that the number of titles and the text content of informational webpages are larger. It is difficult to accurately obtain the title and body content in the project implementation; therefore, the list $pureTextList$ is used to replace the pure text.

$$\begin{aligned} pureTextList = allTextList - scriptTextList \\ -formTextList - optionTextList \\ -inputTextList - hrefTextList \end{aligned} \tag{3}$$

The number of pure texts refers to the character number of the pure texts on the webpage. The calculation of the number $pureTextCount$ of pure texts is shown in Formula 4.

$$pureTextCount = \sum_{i}^{n} \left(pureTextList\left[i\right].length\right) \tag{4}$$

where $pureTextList\left[i\right].length$ represents the number of elements in pure text list $pureTextList$. $\sum_{i}^{len(pureTextList)} \left(pureTextList\left[i\right].length\right)$ represents the cumulative sum of the number of characters of all elements in the pure text list $pureTextList$.

3. Proportion of pure texts

The proportion of pure texts refers to the ratio of the number of pure text characters on a webpage to the number of text characters on the webpage. We believe that there are more pure texts in informational webpages, so the proportion of pure texts is higher. The calculation of the proportion $pureTextPercent$ of pure texts is shown in Formula 5.

$$\begin{aligned} pureTextPercent &= \frac{pureTextCount}{allTextCount} \\ &= \frac{\sum_{i}^{n} \left(pureTextList\left[i\right].length\right)}{\sum_{j}^{n} \left(allTextList\left[j\right].length\right)} \end{aligned} \tag{5}$$

where $\sum_{j}^{n} \left(allTextList\left[j\right].length\right)$ represents the cumulative sum of the number of characters of all elements in the all-text list $allTextList$.

4. Average length of pure texts

The average length of pure texts refers to the average number of characters in the pure text list $pureTextList$ on the webpage. The calculation of the average length $pureTextLength$ of pure texts is shown in Formula 6.

$$\begin{aligned} pureTextLength &= \frac{pureTextCount}{len\left(pureTextList\right)} \\ &= \frac{\sum_{i}^{n} \left(pureTextList\left[i\right].length\right)}{len\left(pureTextList\right)} \end{aligned} \tag{6}$$

We believe that the average length of pure texts of informational webpages is longer because there will be relatively more large sections of texts in the body of the informational webpage.

B. Normalization of feature data

Formulas 2-6 show that among the four feature data items, only the proportion of pure texts is in the range [0,1]. The four feature data items include the number of continuous large texts, the number of pure texts, the proportion of pure texts, and the average length of pure texts. If the value of the feature data item is not mapped to the range [0,1], the KNN algorithm will prefer the feature data items with larger values during training. Therefore, it is necessary to normalize the feature data items.

Taking the number of consecutive large texts as an example, the normalization method is shown in Formula 7.

$$seqentialLargeTextCount = \frac{seqentialLargeTextCount}{max\left(dataListTrained\right)} \tag{7}$$

In formula 7, $dataListTrained$ represents the set of data samples used for training. The denominator is the maximum number of consecutive large texts in $dataListTrained$.

Algorithm 1 : ObtainSequentialLargeTextCount

Input: $pureAndHrefTextList$, $hrefTextList$;
Output: $sequentialLargeTextCount$;
Procedure: ObtainSequentialLargeTextCount($pureAndHrefTextList$, $hrefTextList$);
// The list $isLargeTextList$ is used to mark whether any element in the list $pureAndHrefTextList$ is a large text;
$isLargeTextList$=[];
// The follow programs fill the marks in the list $isLargeTextList$;
for each $currentText$ **in** $pureAndHrefTextList$ **do**
    if $currentText.length \geq 50$ **and** $currentText$ **not in** $hrefTextList$; **then**
        $isLargeTextList$.append(True);
    else
        $isLargeTextList$.append(False);
    end if
end for
// This variable index represents the index of the list $pureAndHrefTextList$;
$index$=0;
// This variable $listLen$ represents the length of the list $pureAndHrefTextList$;
$listLen$=len($pureAndHrefTextList$);
// This variable $sequentialLargeTextCount$ represents the number of consecutive large texts;
$sequentialLargeTextCount$=0;
for each $currentText$ **in** $pureAndHrefTextList$ **do**
    // There is only one element in the list $pureAndHrefTextList$;
    if $listLen \leq 1$ **then**
        break;
    end if
    // The current element is the first element in the list $pureAndHrefTextList$;
    if $index$ == 0 **then**
        if $isLargeTextList$[$index$] **and** $isLargeTextList$[$index$+1] **then**
            $sequentialLargeTextCount$=$sequentialLargeTextCount$+1;
        end if
    end if
    // The current element is the last element in the list $pureAndHrefTextList$;
    if $index$ == $listLen$-1 **then**
        if $isLargeTextList$[$index$] **and** $isLargeTextList$[$index$-1] **then**
            $sequentialLargeTextCount$=$sequentialLargeTextCount$+1;
        end if
    end if
    // The current element is the intermediate element in the list $pureAndHrefTextList$;
    if $index \neq listLen$-1 **and** ($index \neq 0$ **or** $isLargeTextList$[$index$+1]) **then**
        if $isLargeTextList$[$index$] and $isLargeTextList$[$index$-1] **then**
            $sequentialLargeTextCount$=$sequentialLargeTextCount$+1;
        end if
    end if
    $index$=$index$+1;
end for
Return: $sequentialLargeTextCount$;

C. Training and evaluation of KNN model

The result of the KNN model training is the identification of the value of k with the maximum accuracy. Therefore, 20% of the number of experimental data samples can be taken as the maximum value of k. The accuracy can be calculated when the value of k gradually increases.

To obtain stable accuracy, the K-fold method can be used for cross-validation. If the value of K is 10, 90% of the random experimental data samples are used as training data, and the other 10% of the experimental data samples are used as verification data. K experiments are carried out, and the average value of the accuracy of K experiments is taken as the final accuracy. The following Python statements can be used to obtain the accuracy through the K-fold method.

$scores$=$cross\_val\_score$(knn,X,y,$scoring$="$accuracy$", cv=10)

Among these parameters, knn is the KNN model, X is the experimental data, and y is the result data.

The object $cross\_val\_score$ comes from the library Sklearn. We can import this object using the following statement.

from $sklearn.model\_selection$ import $cross\_val\_score$

TABLE II
THE 200 WEBPAGES MARKED IN ADVANCE

website	number of noninformation webpages	number of information webpages	total value
http://www.chinawuliu.com.cn	14	16	30
http://www.gzxdwl.com	86	84	170
The total values	100	100	200

D. Prediction of webpage types

Because the KNN algorithm is a lazy loading algorithm, the KNN model can be generated after obtaining the appropriate value of k. The KNN model can be used for prediction after the model is constructed by loading the experimental data. The KNN model can be generated using the following Python statement.

$knn = neighbors.KNeighborsClassifier(n\_neighbors=k)$

The parameter $n\_neighbors$ is the value of k in the KNN model. We use the following Python statement to load the experimental data.

$$knn.fit(X,y)$$

Next, we develop programs to obtain the values of the four feature data items of the webpages to be classified. We form two-dimensional array data, which are predicted using the following Python statement.

$$knn.predict(datas)$$

VI. CLASSIFICATION EXPERIMENT OF WEBPAGES

The procedure of the webpage classification experiment is described in detail below. Before the experiment, we crawled the website http://www.chinawuliu.com.cn of the CFLP (China Federation of Logistics & Purchasing) and the website http://www.gzxdwl.com of the GZLA (Ganzhou Modern Logistics Association) to obtain 50600 webpages. To facilitate this experiment, 200 webpages were labeled in advance as experimental data.

A. Feature data extraction and annotation results

A summary of the 200 webpages labeled in advance is shown in Table II. The labeled target classification is shown in III. Four feature data were calculated according to Formulas 2-6, and the results are shown in Table III.

The following three points need to be explained in Table III.

1. In the target classifications, 0 indicates that the webpage is a noninformational webpage, and 1 represents that the webpage is an informational webpage.

2. The maximum number of consecutive large texts is 25. The maximum number of pure texts is 8396. The maximum average length of pure texts is 105.857.

3. All data are rounded to three decimal places.

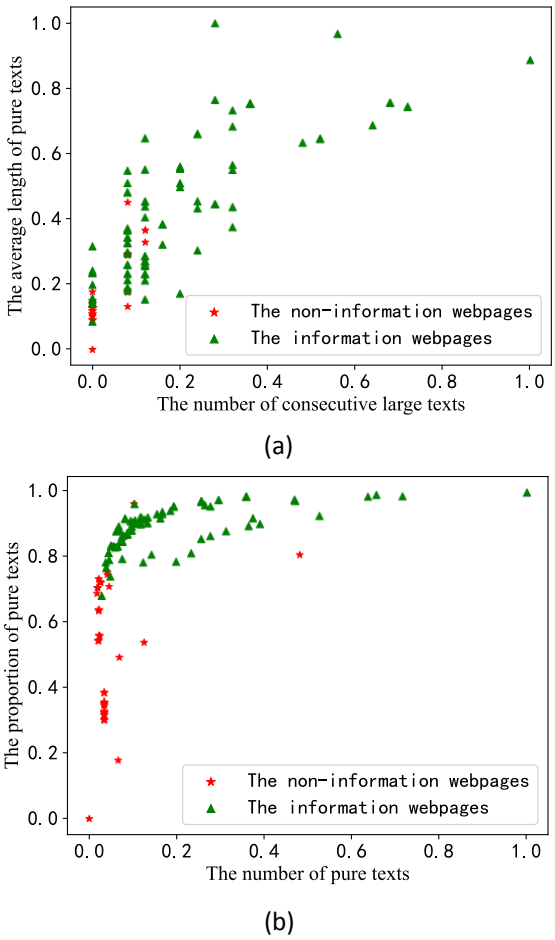

Fig. 3. The scatter diagrams with feature data items

B. Normalization of feature data

We normalize the feature data with reference to Formula 7, and the results are shown in Table IV.

To more intuitively observe whether the feature data items can be used for webpage classification, scatter diagrams are drawn by two feature data items. The number of continuous large texts and the average length of pure texts are shown in Figure 3(a). Most of the distribution areas of the informational and noninformational webpages are obviously different. Similarly, the number of pure texts and the proportion of pure texts are shown in Figure 3(b). It can still be seen that most of the distribution areas of the informational and noninformational webpages are obviously different. However, in both cases, the points of the two types of webpages have a small amount of intersection. Therefore, using these four feature data items can achieve good classification results.

To more intuitively observe whether the feature data items can be used for webpage classification, scatter diagrams are drawn by two feature data items. The number of continuous large texts and the average length of pure texts are shown in Figure 3(a). Most of the distribution areas of the information webpages and the noninformation webpages are obviously different. Similarly, the number of pure texts and the proportion of pure texts are shown in Figure 3(b). It can still be seen that most of the distribution areas of the

TABLE III
THE FEATURE ITEMS OF THE 200 WEBPAGES MARKED IN ADVANCE

sequence number	webpage	number of consecutive large texts	average length of pure texts	Proportion of pure texts	number of pure texts	target classification
1	http://www.gzxdwl.com/Newsshow/id_/VlcxWlRXSkJUbWRXUjBFOQ00_40000400.html	0	15.714	0.719	220	0
2	http://www.china_wuliu.com.cn	2	13.85	0.178	554	0
3	http://www.gzxdwl.com	0	0	0	0	0
4	http://www.chinawuliu.com.cn/_zixun/dfwl/	2	47.388	0.803	4028	0
... ...	... ...	... ...	... ...	... ...	... ...	... ...
200	http://www.chinawuliu.com.cn/_about/contact/	2	19.412	0.766	330	1

TABLE IV
THE FEATURE DATA OBTAINED AFTER NORMALIZATION

sequence number	webpage	number of consecutive large texts	average length of pure texts	proportion of pure texts	number of pure texts
1	http://www.gzxdwl.com/Newsshow/id_/VlcxWlRXSkJUbWRXUjBFOQ00_40000400.html	0	0.148	0.719	0.026
2	http://www.china_wuliu.com.cn	0.08	0.131	0.178	0.066
3	http://www.gzxdwl.com	0	0	0	0
4	http://www.chinawuliu.com.cn/_zixun/dfwl/	0.08	0.448	0.803	0.480
... ...	... ...	... ...	... ...	... ...	... ...
200	http://www.chinawuliu.com.cn/_about/contact/	0.08	0.183	0.766	0.039

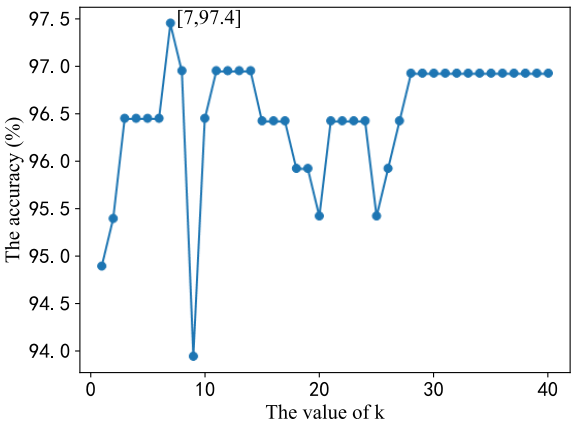

Fig. 4. Webpage information recognition accuracy for different k

information webpages and the noninformation webpages are obviously different. However, in both cases, the points of the two types of webpages have a small amount of intersection. Therefore, using these four feature data items can achieve good classification results.

C. Training and evaluation of KNN model

After training and using K-fold cross-validation, the accuracy of taking different k was obtained and is shown in Figure 4. The highest accuracy of 97.5% was achieved when k was 7.

D. Prediction of webpage types

Next, the values of the feature data were obtained from the webpages. The webpages were predicted by the same process of feature data extraction and normalization. After the feature data matrix was obtained, the classification results of webpages were obtained by the function $knn.predict()$. Finally, the experiment completed the classification of 50660 webpages and identified 38035 informational webpages.

VII. CONCLUSION

Using four steps, we construct the KNN model for the dichotomous classification of webpages. The four steps are extraction and annotation of feature data, normalization of feature data, training and evaluation of the KNN model, and prediction of webpage types. We can identify the informational webpages by using the KNN model.

The main points of the method for identifying informational webpages are to understand the business scenario, extract appropriate feature data for classification, and then determine the value of k with the maximum accuracy. Experiments show that using the four feature data items results in a good dichotomous classification ability. The four items are the number of continuous large texts, the number of pure texts, the proportion of pure texts, and the average length of pure texts. The accuracy reaches 97.5% of the peak when the value of k is 7. The method for identifying informational webpages has engineered practical value. Therefore, we use the KNN model to identify 38035 informational webpages on 50660 webpages, and lay a data foundation for subsequent natural language processing tasks.

REFERENCES

[1] M. R. Murugudu and L. Reddy, "Efficiently harvesting deep web interfaces based on adaptive learning using two-phase data crawler framework," *Soft Computing*, vol. 10, no. 7, pp. 1–11, 2021.

[2] B. Tidke, R. Mehta, D. Rana, and H. Jangir, "Topic sensitive user clustering using sentiment score and similarity measures: big data and social network," *International Journal of Web-Based Learning and Teaching Technologies (IJWLTT)*, vol. 15, no. 2, pp. 34–45, 2020.

[3] S. Anupam and A. K. Kar, "Phishing website detection using support vector machines and nature-inspired optimization algorithms," *Telecommunication Systems*, vol. 76, no. 1, pp. 17–32, 2021.
[4] P. O. L. Junior, L. G. de Castro Junior, and A. L. Zambalde, "Analysis of machine learning techniques to classify news for information management in coffee market," *IEEE Latin America Transactions*, vol. 13, no. 7, pp. 2285–2291, 2015.
[5] Y. Zhong, L. Luo, X. Wang, and J. Yang, "Multi-factor stock selection model based on machine learning," *Engineering Letters*, vol. 29, no. 1, pp. 177–182, 2020.
[6] H. Huang, Y. Gao, H. Zhang, and B. Li, "Weighted lasso estimates for sparse logistic regression: non-asymptotic properties with measurement errors," *Acta Mathematica Scientia*, vol. 41, no. 1, pp. 207–230, 2021.
[7] M. Agarwal, K. K. Rao, K. Vaidya, and S. Bhattacharya, "Ml-moc: Machine learning (knn and gmm) based membership determination for open clusters," *Monthly Notices of the Royal Astronomical Society*, vol. 502, no. 2, pp. 2582–2599, 2021.
[8] W. Zhu, H. Wang, and X. Zhang, "Synergy evaluation model of container multimodal transport based on bp neural network," *Neural Computing and Applications*, vol. 33, no. 9, pp. 4087–4095, 2021.
[9] D. Gamdha, S. Unnikrishnakurup, K. J. Rose, M. Surekha, P. Purushothaman, B. Ghose, and K. Balasubramaniam, "Automated defect recognition on x-ray radiographs of solid propellant using deep learning based on convolutional neural networks," *Journal of Nondestructive Evaluation*, vol. 40, no. 1, pp. 1–13, 2021.
[10] K. ElDahshan, E. Elsayed, and H. Mancy, "Enhancement semantic prediction big data method for covid-19: Onto-nosql," *IAENG International Journal of Computer Science*, vol. 47, no. 4, pp. 613–622, 2020.
[11] X. Chen, Y. Yang, S. Wang, H. Wu, J. Tang, J. Zhao, and Z. Wang, "Ship type recognition via a coarse-to-fine cascaded convolution neural network," *The Journal of Navigation*, vol. 73, no. 4, pp. 813–832, 2020.
[12] J. Schaefferkoetter, J. Yan, C. Ortega, A. Sertic, E. Lechtman, Y. Eshet, U. Metser, and P. Veit-Haibach, "Convolutional neural networks for improving image quality with noisy pet data," *EJNMMI research*, vol. 10, no. 1, pp. 1–11, 2020.
[13] H. Lou and M. J. Hageman, "Machine learning attempts for predicting human subcutaneous bioavailability of monoclonal antibodies," *Pharmaceutical Research*, vol. 38, no. 3, pp. 451–460, 2021.
[14] X. Shi, Z. Guo, F. Xing, Y. Liang, and L. Yang, "Anchor-based self-ensembling for semi-supervised deep pairwise hashing," *International Journal of Computer Vision*, vol. 128, no. 8, pp. 2307–2324, 2020.
[15] R. C. Guido, "Paraconsistent feature engineering [lecture notes]," *IEEE Signal Processing Magazine*, vol. 36, no. 1, pp. 154–158, 2018.
[16] Z. Deng, T. He, W. Ding, and Z. Cao, "A multimodel fusion engine for filtering webpages," *IEEE Access*, vol. 6, no. 1, pp. 66 062–66 071, 2018.
[17] B. Al-Helali, Q. Chen, B. Xue, and M. Zhang, "A new imputation method based on genetic programming and weighted knn for symbolic regression with incomplete data," *Soft Computing*, vol. 25, no. 8, pp. 5993–6012, 2021.
[18] L. Wuke, Y. Guangluan, and C. Xiaoxiao, "Application of deep extreme learning machine in network intrusion detection systems," *IAENG International Journal of Computer Science*, vol. 47, no. 2, pp. 136–143, 2020.
[19] Y. Zhao, G. Deng, L. Zhang, N. Di, X. Jiang, and Z. Li, "Based investigate of beehive sound to detect air pollutants by machine learning," *Ecological Informatics*, vol. 61, no. 1, pp. 101 246–101 260, 2021.
[20] G. F. Simiqueli and M. D. V. de Resende, "Entropy and mutual information in genome-wide selection: the splitting of k-fold cross-validation sets and implications for tree breeding," *Tree Genetics & Genomes*, vol. 16, no. 2, pp. 1–14, 2020.