Research on Text Classification Based on TextRank

Guangming Lu[1,2], Yule Xia[3], Jiamei Wang[1,2,*] and Zhenling Yang[1]
[1]School of Electrical Information Engineering, Yunnan Minzu University, Kunming, Yunnan, 650500, China
[2]Yunnan Province College minority language information processing engineering research center, Kunming, Yunnan, China
[3]Changsha University of Science & Technology, Changsha, Hunan, 410114, China
[*]Corresponding author

***Abstract*—Extracting keywords from the result of word segmentation with the improved TextRank algorithm. Use the relative position of the words in the article to calculate the influence of position; the position of the coverage of the words and expressions is extended to the statement of the words and the key words as the feature of the text. Hadoop programming using naive Bayesian algorithm for text classification. The experiments show that the improved Textrank has a great improvement in classification performance, and the classification accuracy of naive Bayesian algorithm is 93% when the number of keywords is 40. Compared with the traditional, the accuracy rate increased by about 10%.**

Keywords-component; hadoop; TexTrank; naive bayes; text classification

I. INTRODUCTION

In recent years, the keyword extraction method based on graph model has received extensive attention. By PageRank [5] in the information retrieval field achieved great success of enlightenment and reference [8] proposed a TextRank graph model based on. Among them, the nodes of a graph is the candidate words and edges reflect the co-occurrence relations between words in documents and words. Based on the idea that similar to TextRank, PageRank thinks that a word that has a strong connection with many important words is an important word. Use PageRank algorithm to calculate the importance of words in the figure, and then according to the PageRank score of the candidate key words to sort, so as to choose the highest number of words as the key words. [7] also will be the same as the first application in the [6] page ranking HITS algorithm for the candidate key words, in the keyword extraction performance on the HITS algorithm and TextRank performance is similar.

Later, there are a lot of research is to improve the classical TextRank algorithm. Wan Xiaojun [11] presents a graph model of the algorithm can also extract keywords and key sentences, between the word and the word. In other words and have established a link. This method can make full use of the structure of the network, is a strengthened version of the TextRank algorithm, and achieved good results. Next, million [9, 10] and through a lot of topics related to the topic of the document and the document to solve the problem of limited resources in a single document. Experiments show that the improved version of the TextRank effect is better than the original TextRank method.

Liu [12] proposed a specific topic of the TextRank algorithm, they believe that the importance of words should be dependent on a particular topic, and a good topic model is introduced into the graph model. Experiments show that their method is superior to many classical methods, including classical TextRank. Reference [12] to continue to improve the work of reference [13], they believe that in addition to the importance of the words on the map (node) depends on the topic, the degree of common words (edge) is also dependent on the topic. An improved version of the TextRank model is proposed, and a discriminative model is proposed based on the characteristics of the topic keywords on micro-blog. Experiments on micro-blog show the effectiveness of their methods.

Above reference only for keywords extraction were studied, and has achieved good results, but did not keywords applied to the text classification, in text classification, text feature extraction is a crucial step, people's traditional feature extraction has been done a lot of research, but in the classification results above has no room for improvement. And with the rapid development of the Internet, text documents on the Internet is also a dramatic increase in the, these documents including research reports, academic papers, news, online data bank, how to effectively and quickly to deal with the text information, from the mass of information isolated valuable information is a hotspot of current research. Traditional text classification technology has reached a very high level, however, with the increase of network information, the traditional technology has been difficult to deal with a large number of text messages.

This paper will combine the TextRank, Bias and Hadoop to deal with the massive text data. First application the TextRank algorithm for text keywords extraction, from keywords to extract text features, then use Bayesian after line of text classification. Finally, the above algorithms written for MapReduce program, and the initial data is uploaded to the HDFS, in a cluster of calculation and test, and find out the different themes under, the number of keywords to the text classification result.

II. IMPROVED TEXTRANK AND TEXT KEYWORD EXTRACTION

A. Construction of Candidate Key Words

Split the text into sentences, filter out the stop words in each sentence, and just keep the word of the specified part of speech. From this, we can get the set of sentences and the set of

words. Each word as a node in TextRank. Set the window size to k, assuming that a sentence is followed by the following words: $x_1, x_2, \dots, x_k, \dots, x_n$ .Among them $(x_1, x_2, \dots, x_k), (x_2, x_3, \dots, x_{k+1}), (x_3, x_4, \dots, x_{k+2})$ is a window. There is a non - right side between the nodes of the two words in a window. Based on the above chart, you can calculate the importance of each word node. The most important words can be used as key words. Algorithm is as follows:

$$W(x_i) = (1-d) + d * \sum_{x_j \in PoinIn(x_i)} \frac{w_{ji}}{\sum_{V_k \in PointOut(x_j)} w_{jk}} W(x_j) \quad (1)$$

$W(x_i)$ is the importance of the word x_i , Similarly, $PointIn(x_i)$ is a collection of words that point to the word x_i, $PointOut(x_j)$ for the word x_j to point to the collection of other words. w_{ji} is the strength of the connection between x_i and x_j (the influence of words). d is the damping coefficient, generally set to 0.85.

Is the main contribution of TextRank PageRank idea is introduced into the text composition unit importance ranking field, TextRank algorithm is applied to keyword extraction and construction is an undirected unweighted graph, each node is given an initial value of 1, then iterative calculation weights. It seems intuitive, can according to some strategy for an important part of the node given a higher initial value, in order to improve the ranking results, but this method does not work. As a matter of fact, to sort the results corresponding eigenvectors of the matrix transfer, given by the nodes and independent of initial value, but are connected by the weight of the edge node of the decided. Based on the candidate key words, we discuss how to introduce the weight of the edge in order to improve the ranking effect.

B. Text Keyword Extraction

In the same piece of articles, most of the words are related to the article, there is a definite relation between these words. The stronger correlation with themes of words, will frequently appear in these connections. So, the voting mechanism between web pages into voting between words, we can find the key words. TextRank is based on the graph sorting algorithm, and for undirected graph, there is no difference between the edge of the. Reference [14], in the weight assignment and keyword extraction, considering the word coverage importance, importance of the position, and the frequency of the importance and the combination of the three as the weight of words, however, the author in the calculation of the coverage of the importance is will the influence of words are uniform transfer to neighboring nodes, without considering the position of the words the importance; calculated position importance and considers only the two locations: the title and the main body of the paper. The change algorithm was improved, will cover the importance and the importance of location into consideration and give priority to the importance of the position, because of an article summary paragraphs usually headed at the end of the paragraph, the important sentences appear in the first paragraph at the end of the sentence. In the reference [14], the proportion of the three influence to a word is expressed as α、β、γ, and $\alpha + \beta + \gamma = 1$.

For any two nodes v_i and v_j. The influence of node v_i on v_j is transmitted through its directed edge $e = < v_i, v_j >$. The weight of the edge determines the v_j to get the v_i part of the value of the size, the three influence of the improved algorithm is as follows:

1. The $w\alpha(v_i, v_j)$ indicates that the coverage of v_i is transferred to the weight of v_j , and the calculation is as follows:

$$w\alpha(v_i, v_j) = \frac{\lambda}{|PointOut(v_i)|} \quad (2)$$

$$\lambda = \frac{1}{k} \quad (3)$$

Which $PointOut(v_i)$ is the number of nodes around the node v_i, v_i around the nodes only where the sentences; k is the distance around the node and the node v_i, k is greater than or equal to 1, neighboring nodes, $k = 1$, the farther the distance, transferring the weight of smaller and smaller.

2. The $w\beta(v_i, v_j)$ indicates the location of v_i transfer to influence the weight of v_j calculated as follows:

$$w\beta(v_i, v_j) = \frac{I(v_j)}{\sum_{v_k \in PointOut(v_i)} I(v_k)} \quad (4)$$

$$I(v) = \frac{\left|\frac{WordId(v)}{N} - 0.5\right|}{0.5} \quad (5)$$

3. The $I(v)$ said the word v the importance value, $WordId(v)$ for the location of the word v in the text, the value is greater, said after the word position by, N is the text of the total number of words; by formula (5) shows, closer to the words at the beginning and end of the more important.

4. The $w\gamma(v_i, v_j)$ gamma said v_i frequency influence passed to the weight of v_j, calculation method is similar with reference [14] , not repeat them here.

By the above *calculation*, we can get the formula (1) in the w_{ij}:

$$w_{ij} = \alpha * w\alpha(v_i, v_j) + \beta * w\beta(v_i, v_j) + \gamma * w\gamma(v_i, v_j) \quad (6)$$

When uses the TextRank algorithm to calculate the fraction of nodes in the graph and need to to a node in the graph specify arbitrary initial values and recursive calculation until a word score convergence, convergence after each node are obtained a score, on behalf of the nodes in the graph the importance of. And the final score of the node is not affected by the given initial value, the initial value only affects the number of iterations to achieve convergence. Based on the above chart, you can calculate the importance of each word node. The most important words can be used as key words.

III. BIAS TEXT CLASSIFIER

In the field of data mining, Bayesian classification established in probability and statistics theory, in a variety of classification problems has been widely used, compared to other classification methods, and its uniqueness is not absolute distribution of an object to a class, but gives the object can be can be attributed the probability of each category, then according to the size of the probability to determine the categories.

Naive Bayesian classifier is a simple classification algorithm, called the naive Bayesian classification is because the thought of this method is really very simple, the ideological foundation of the naive Bayes is such: the are to be classified items, probability of each category in solution in the condition, which is the biggest, that this to be classified items which belong to the category.

The steps of the Bias classification are as follows:

1. Set $\boldsymbol{X} = \{x_1, x_2, \dots, x_m\}$ is a pending category, and each x is a feature attribute of $\boldsymbol{X}$ (key words);

2. Collection of all categories $\boldsymbol{Y} = \{y_1, y_2, \dots, y_n\}$

3. Calculation of $P(y_1|\boldsymbol{X}), P(y_2|\boldsymbol{X}), \dots, P(y_n|\boldsymbol{X})$;

4. If $P(y_k|\boldsymbol{X}) = max\{P(y_1|\boldsymbol{X}), P(y_2|\boldsymbol{X}), \dots, P(y_n|\boldsymbol{X})\}$, then $\boldsymbol{X} \in y_k$.

The key of classification is to calculate the probability of each condition in the first *3)* step, and the method is as follows:

Find a set of known categories to be classified, which is called a training sample set. Conditional probability estimates for each feature attribute under various categories are obtained. That is:

$$\begin{matrix} P(x_1|y_1), P(x_2|y_1), \dots, P(x_m|y_1) \\ P(x_1|y_2) P(x_2|y_2), \dots, P(x_m|y_2) \\ \vdots \\ P(x_1|y_n), P(x_2|y_n), \dots, P(x_m|y_n) \end{matrix} \quad (7)$$

If each characteristic attribute is independent of the condition, the derivation is based on Bias's theorem:

Because the denominator for all categories is constant, because as long as we can maximize the molecular. And because the attributes are independent of the conditions, there are:

$$P(X|y_i)P(y_i) = P(x_1|y_i)P(x_2|y_i) \dots P(x_m|y_i) = P(y_i) \prod_{j=1}^{m} P(x_j|y_i) \quad (8)$$

IV. EXPERIMENTAL PROCESS AND RESULT ANALYSIS

This experiment uses the news corpus, which involves the economy, science and technology, medical, sports and other 9 categories, each class of 2000 documents, a total of 18000 text files. First of all these documents, the word segmentation, word processing, and then use the TextRank algorithm to extract the key words in the document, and the number of keywords extracted on the impact of the classification results were analyzed.

A. Training and Testing of Naive Bias Classifier

From the segmentation results we know that samples in each row corresponds to a key-value (categories - keywords) and we will upload files to a cluster, with pig the segmentation ratio: 80% for training samples, 20% as test samples, then use mahout's command *trainclassifier* training, testing with the *testclassifier*, training and testing, including following several MapReduce process [2]:

1. Collection of features, and produce a feature dictionary, fixed feature words set;

2. Calculate the total number of all kinds of text;

3.The total number of words of each class are calculated;

4. Calculated for each feature words of each text belongs to the probability of each class.

5. Classification.

B. Evaluating Indicator

We usually evaluate the classification results of each category will be used to recall and precision, the method is as follows:

$$R(y_i) = \frac{TP_i}{TP_i + FN_i} \quad P(y_i) = \frac{TP_i}{TP_i + FP_i} \quad (9)$$

The TP to judge belongs to a category and practical is the category of text, FN for judgment belongs to a category rather than a reality of the class text number FP for judgment does not belong to a class but is actually the category number of texts.

However, each type of recall and precision are evaluation model on the performance of a certain class represent only local significance, all we when we want to have the performance of the overall evaluation model, you need to use macro averaging and F value:

$$MR = \frac{\sum_{i=1}^{n} R(y_i)}{n} \quad (10)$$

$$MP = \frac{\sum_{i=1}^{n} P(y_i)}{n} \quad (11)$$

$$F = \frac{2 \times MP \times MR}{MP + MR} \quad (12)$$

which n represents the total number of categories.

C. Experimental Results and Analysis

We are setting $\alpha = \beta = \gamma = 1/3$. In order to analyze the influence of the number of keywords on the classification results, the key words extracted from the 5 start, gradually increased to 50 key words. At the same time, the speed of the training and testing of the classification is done.

From Table I can see, when the keyword take five, quite on each article only removed five words as features, F value can

reach more than 0.9, indicating that the algorithm the extracted keyword is very accurate, can truly reflect the characteristics of; when the number of keywords to 40, and F value reaches the maximum 0.933797. Compared with the traditional naive Bayes text classification algorithm [15] to enhance the 10%, which fully shows that using the key words as text features than the traditional feature selection methods have greatly improved.

TABLE I. RESULT

Keyword Num	MR	MP	F
5	0.909787	0.909748	0.909767
10	0.923163	0.922205	0.922684
15	0.924138	0.923390	0.923764
20	0.931081	0.930734	0.930907
25	0.925026	0.924024	0.924525
30	0.9297674	0.929786	0.929845
35	0.929282	0.926464	0.927871
40	0.934438	0.933155	0.933797
45	0.929869	0.927397	0.928632
50	0.927142	0.925694	0.926418

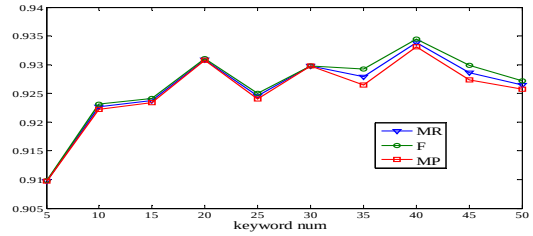

FIGURE I. THE PERFORMANCE OF THE CLASSIFICATION OF THE NUMBER OF KEY WORDS CHANGE

As can be seen from Figure Ⅰ, the classification performance in the process of increasing the key words, which may be affected by the performance of the virtual machine, but the overall performance is better. When the number of words is a 40, classification results better, and when the number of keywords to increase, classification performance began to decline and the efficiency will decline, figure 5 for comparison between the training and testing time in the Hadoop cluster.

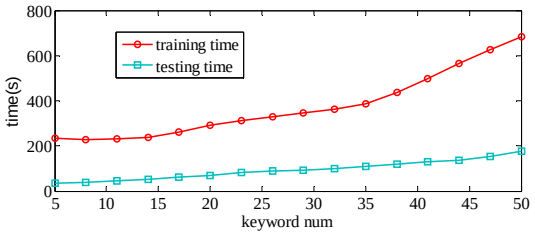

FIGURE II. THE PERFORMANCE OF THE CLASSIFICATION OF THE NUMBER OF KEY WORDS CHANGE

As can be seen from Figure II, with the increase of the number of keywords, the training time is also a sharp increase, combined with figure 4, this time the classification performance also began to decline.

V. CONCLUSION

This paper combines the TextRank algorithm and the naive Bayes algorithm on the cloud computing platform Hadoop, and carries on the text classification technology research. The weight algorithm proposed by [14] is improved, and the key words are used as text features. Experimental results show that compared with the traditional algorithm, the text classification efficiency and accuracy are greatly improved when the extraction keywords are used as features.

ACKNOWLEDGMENT

This work was financially supported by the National Natural Science Foundation of China (61363085), Major projects in Yunnan Province (ZD2013013), High level construction of university scientific research project of Yunnan MinZu University. State Language Commission authorize research projects (WT125-61).

REFERENCES

[1] Rada Mihalcea and Paul Tarau. TextRank: Bringing Order into Texts.In Proceedings of Empirical Methods in Natural Language Processing 2004.

[2] Li BinBin. Research of Text Classification Based on Hadoop,2015

[3] Sean Owen, Robin Anil, Ted Dunning. Mahout in Action, 2010:274-280.

[4] Yang J, Ji D, Cai DF. Keyword Extraet1on In Mu1ti-Doeument Based on TextRank Teehno1ogy. NCIRCS,2008

[5] Sergey Brin Lawrence Page. The anatomy of a large-scale hypertextual web ,search engine. In Proceedings of WWW pages 107-117, 1998.

[6] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. In Proceedings of ACM-SIAM symposium on Discrete algorithms, 1998.

[7] Litvak M and Last M. Graph-based keyword extraction for single-document summarization. In Proceedings of Workshop Multi-source Multilingual Information Extraction and Summarization,2008.

[8] R. Mihalcea and P. Tarau. TextRank: Bringing order into texts. In Proceedings of EMNLP, 2004.

[9] Xiaojun Wan and Jiangguo Xiao. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In Proceedings of International Conference on Computational Linguistics(COLING2008) pages 969-976, 2008.

[10] Xiaojun Wan and Jianguo Xiao. Single document keyphrase extraction using neighborhood knowledge. In Proceedings of AAAI.2008.

[11] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In Proceedings of ACL, 2007.

[12] Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. Automatic keyphrase extraction via topic decomposition. In Proceedings of Conference on Empirical Methods in Natural Language Proceeding, pages 366-376, 2010.

[13] Wayne Xin Zhao, Jing Jiang, Jing He, Yang Song, Palakorn Achananuparp, EePeng Lim, and Xiaoming Li. Topical keyphrase extraction from twitter. In Proceedings of ACL, 2011.

[14] Xia Tian .Study on Keyword Extraction Using Word Position Weighted TextRank.New Technology of Library and Information Service.2013

[15] LI Jing-mei, SUN Li-hua, ZHANG Qiao-rong, ZHANG Chun-sheng.Application of native Bayes classifier to text classification.Journal of Harbin Engineering University.2003