Review of the Classification of Massive Chinese Texts Based on Spark

Liu Yu [1,a]

[1] School of Software Technology, Zhejiang University, Zhejiang, China

Abstract. As the Internet develops rapidly, the number of texts is also growing rapidly. Whether it is the content of online emails exchanged by people, or the online novels and other literary contents, or news reports, personal blogs, Weibo or comments, they are constantly increasing the amount of text at all times. However, most of the data is not classified or processed, which causes a lot of spam, junk information, meaningless articles or advertisements. Their production not only consumes a lot of Internet resources, but also affects users' online experience and reduces the users' work and study efficiency. Therefore, it is vital accurately classify a large amount of text, judge its nature according to the classification result, and carry out targeted treatment. The classification of massive texts based on Spark framework is reviewed in this paper.

1 Introduction

With the development of Internet technology and social media, massive network text data has been derived. However, most of the massive data has not been processed and classified, which results the emergence of bad network behaviors, such as spam and advertisement push, making it difficult for people to extract useful information from massive data, which wastes a lot of users' time and energy to process spam. Therefore, how to efficiently classify massive text data has important theoretical significance and application value [1], and how to efficiently extract valuable information in massive text information has become a research hot spot [2].

Text classification technology, as a key technology for text processing, has been widely used in improving information retrieval and utilization [3]. At present, the classification algorithms, such as K-neading algorithm [4], Naive Bayes [5], Maximum Entropy [6], Support Vector Machine (SVM) [7], Artificial Neural Network [8], decision tree [9], and rough sets [10], are widely used in practical applications. The researches on the above text classification methods focus on the small and medium-scale data model. The traditional classification system seems to be powerless for the large-scale data and the scenes that require higher real-time performance. Therefore, many predecessors have organically combined the framework of big data with traditional machine learning, in order to solve the problem that traditional text classification can not complete the classification of massive texts [11].

The MapReduce framework is the most widely used big data parallel computing framework. People have attached more attention to the research on parallel text classification algorithms under the MapReduce framework. The disadvantage of the MapReduce framework is that it stores intermediate results on HDFS during parallel computing, leading to a large amount of IO overhead. While the Spark framework is a parallel framework based on memory computing, and it does not directly store the intermediate results on the disk during the performance process (the data portion is cached to disk only when the memory is insufficient), so the performance efficiency of Spark framework is relatively good [12].

2 Current situation of text classification

Bayes classifier is the most classic method in the study of classifiers. It has been widely used in many scenarios, for example, it is found that if the Bayes classifier is used for spam filtering, the phrasal characteristics and other attribute features can improve the classification accuracy rate, up to 95% [13]. Some studies use language model [14] to estimate the correlation probabilities, and have achieved good results. In addition, the online Bayes method has also been widely used in text classification and information filtering [15]. However, the Bayes method requires that the vocabulary between documents should be independent with each other. This conditionally independent hypothesis is not easy to apply to the actual text, so there is often a gap between the actual effect and the theoretical valuation.

Support vector machine is another excellent classification algorithm. It is developed based on a statistical model. The method is based on the statistical VC dimension theory and structural risk minimization principle. It has many advantages in solving small sample, nonlinear and high-dimensional pattern recognition problems, and has been used in pattern

[a] Corresponding author: service@52exe.cn

recognition, regression estimation, probability density function estimation and so on. In the field of text classification, support vector machine classifier has better classification performance and generalization ability, and is widely used in classification field [16-21]. The Liblinear classifier based on the theoretical design of quadratic soft-interval support vector machine designed by Lin Zhiren et. al. of Taiwan University has effectively solved the classification speed problem of traditional support vector machine classifiers, and further accelerates the application of support vector machines in the field of text classification. However, when the amount of data is large, the high-dimensional sparse text vectors bring higher VC dimensions, which will increase the gap between the expected risk and the empirical risk of the classifier, and reduce the performance and accuracy of the classification.

Recently, research on deep learning has received great attention. The deep learning algorithms have achieved amazing results in the field of image recognition and speech recognition [22] [23]. For text data, it is mainly used in natural language processing and semantic mining, such as the presenting of algorithms of word vector, convolutional neural network (CNN) [24] [25], and recurrent neural network (RNN). The idea of training a language model with a neural network was first proposed by Xu Wei of Baidu IDL in 2000 [26][27]. The paper proposed a method for constructing a binary language model using neural network. Subsequently, Bengio et al. published an article on NIPS [28] to give a classical algorithm for training language model with neural network. Then, in the field of NLP, the neural network enters a rapid development period, until recently the results of deep learning once again create a huge wave. Usually, the task of text is done with CNN, its unique convolution, pooling structure can extract the structure, and finally combine the fully connected network to achieve the aggregation and output of information. While RNN provides a way to handle the context in NLP for its memory function.

However, in the field of text classification, the convolution neural network and word vector based on semantics have not achieved theoretical results. The reason is mainly because that deep neural network consumes a lot of computation in semantic recognition, while the semantic quasi-region can not bring the accuracy of classification, instead, the accumulation of defective products and the traditional BOW mode can improve the accuracy of classification. In addition, because the amount of computation of deep learning is large, it is usually necessary to use clusters to calculate, so the computing power of distributed computing engines and clusters is also one factor that affects the classification performance [29].

3 PREPROCESSING OF TEXT

The preprocessing of text mainly includes text formatting, word segmentation, and removing the stop words and other operations. After the text is merged, the entire training set is merged into one file, one line representing a text. The operations of text formatting, word segmentation, and removing the stop words are conducted with the text as a object, so the text preprocessing module has natural parallelism [30]. Currently, the commonly used Chinese word segmentation tools include the word segmentation system ICTCLAS of Chinese Academy of Sciences [31], IKAnalyzer [32] and Paoding [33]. The word segmentation system ICTCLAS of Chinese Academy of Sciences is adopted in this paper, which uses the N shortest path algorithm. ICTCLAS won the first place in the 973 evaluation in 2002, as shown in Table 1 [34-38].

Table 1. Test Results of ICTCLAS in 973 Evaluation

Field	Number	SEG	TAGI	PRAG
Physical Education	33,348	97.01%	86.77%	89.31%
International	59,683	97.51%	88.55%	90.78%
Literature and Art	20,524	96.40%	87.47%	90.59%
Legal Institutions	14,668	98.44%	85.26%	86.59%
Theory	55,225	98.12%	87.29%	88.91%
Economics	24,765	97.80%	86.25%	88.16%
Total	208,213	97.58%	87.32%	89.42%

Before preprocessing, it is necessary to merge the corpus and respectively merge the training set text and the test set text into one text, and each line represents a record. The corpus text content needs to be formatted during the process of merging files. The merged text is uploaded to the HDFS distributed file system as an input file. While segmenting each record, we use the stop word list to remove the word in the text. The word after the word segmentation has a single word and phrase. This is selected as the feature item in this paper, so in the preprocessing, the single word obtained after removing the participle is needed[39-41]. Text preprocessing is implemented on a Spark cluster, and the data items in the RDD are the content of each line in this text. The text content is divided, and the stop words are removed, and the term frequency forms a property dictionary. These distributed operations are performedon the Worker node [42]. The execution flow of preprocessing under Spark is shown in Figure 1.

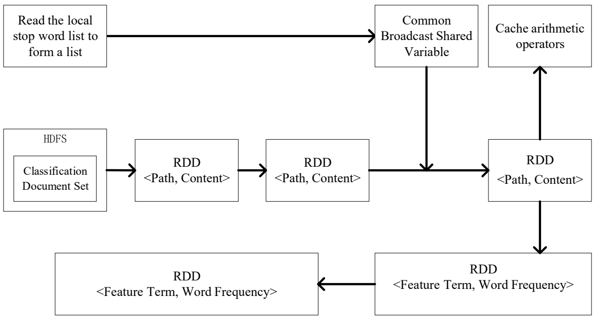
Read the local stop word list to form a list
Common Broadcast Shared Variable
Cache arithmetic operators
HDFS
Classification Document Set
RDD <Path, Content>
RDD <Path, Content>
RDD <Path, Content>
RDD <Feature Term, Word Frequency>
RDD <Feature Term, Word Frequency>

Figure 1. Preprocessing Structure Diagram of Spark Text Classification

4 Text vectorization

The common text vectorization algorithms include word frequency statistics technology [43] [44], TF-IDF algorithm

[45] [46], LDA [47] [48], and Word2vec [49] [50] [51]. The TF-IDF algorithm is the most common algorithm, which combines the Spark.

Li Tao et al. proposed the calculation process of improved feature weighting algorithm in the Spark in the article "Study on Efficient Web Text Classification System under Spark Platform". The classic TFIDF weight calculation is very difficult in the calculation of massive text classification, which takes dozens of hours or even days. This is obviously an unimaginable disaster for occasions with high real-time performance. Therefore, the distributed computing model Spark based on memory is introduced:

Luo Yuanshuai also used TF-IDF as a word vector algorithm combined with Spark in the article "Study on Parallel Text Classification Algorithm Based on Random Forest and Spark".

And he proposed the process: A complete text vectorization process first reads the text RDD fenci after the word segmentation, then uses the feature lexicon RDD features to filter the text content, and counts the TF value of the words in the feature lexicon, and counts the IDF value based on this. After the statistics of TF and IDF are completed, the text is vectorized combining the feature lexicon RDD features to obtain the vector space model RDD TF-IDF[38].

5 Text classification algorithm

Currently, the common text classification methods are: K proximity algorithm[4], Naive Bayes [5], maximum entropy [6], support vector machine (SVM) [7], artificial neural network [8], decision tree[9] , random forests, rough sets [10], etc., but only the KNN and random forests combine Spark and there are related theoretical researches about this.

Yu Pingping et. al. have published "Efficient KNN Chinese Text Classification Algorithm Base on Spar". In this paper, the authors show that it will reduce the classification accuracy in the general parallelization process of KNN text classification, so when using, it is necessary to introduce the relevance of words in the process of calculating the similarity between the training samples and the samples to be tested, improve the classification accuracy and achieve parallelization under the Spark calculation framework, and reduce the computation time [3].

Yan Jiaming et al. proposed a weighted naive Bayes algorithm in the article "Research and Application of Text Classification Based on Cloud Computing" [52], which is an improvement on the naive Bayes algorithm. On the basis of this, it is further improved to the weighted naive Bayes algorithm based on cosine similarity in this paper, and the over-dependence of traditional naive Bayes on the independence of conditional hypothesis is further improved. For example, when the predetermined assumption cannot be satisfied in the actual processing, the classification effect will also be reduced under the premise that the processing data attribute correlation is large. In the classification, if the distribution of the training set can not reflect the distribution of all the data, then the credibility of the prior probability obtained in the middle is not very accurate, its accuracy is needed to be improved through adjustment.

In addition, Luo Yuanshuai et al. proposed the combination of random forest and Spark in "Study on Parallel Text Classification Algorithm Based on Random Forest and Spark", and then realized the classification of massive Chinese texts. Ren Yitian et al. published "Study on the Parallelization Technology of Massive Text Classification Based on Support Vector Machine", which combines SVM algorithm with Spark to classify massive texts.

It requires good RDD to parallelize SVMs with Apache Spark. The RDD is generated from the input of the training data set, and multiple transformations and buffers are performed in the later calculations. Through the transformation, the intermediate variables of the size of the SVM model and the the data grouped by label can be calculated.

It can be found from the above that under the Spark framework, the traditional text classification algorithm is usually simply adjusted or optimized, running on a distributed cluster, so that the traditional processing efficiency can be greatly improved, which has a major contribution to production, life and learning[52][53].

6 Conclusion

With the rapid development of the Internet, text classification technology has also been driven by demand, and has been greatly developed. Currently, there are many mature algorithms that are mature or have been verified, but the classification of massive Chinese texts is still in the stage of development. When classifying, word segmentation may lead to the semantic deviation of articles and error in classification. The incorrect classification will be caused by the inaccurate keyword extraction due to the incorporation of some English words. It is also possible that because of a textual description it may associate with multiple classes. At this time, it is difficult to classify it explicitly. With the emergence of massive texts, text categorization technology is particularly important. Through the reviews on massive Chinese text classification based on Spark technology by predecessors , it is found that although some people have been moving forward in this direction, and it is difficult to improve the accuracy and efficiency based on massive data processing. Therefore, it is still worth studying to conduct segmentation, vectorization and classification of massive texts based on the Spark framework.

References

1. Song Fuxing.Design and Implementation of Spark-Based Super Large Text Classification method [D]. Beijing Jiaotong University, 2017.
2. Wang Xiaolin, Lu Luoyong, Shao Weipeng. Research on Similarity Algorithm of New Words

Based on Information Entropy[J]. Computer Technology and Development, 2015(9): 119-122.

3. Lu L R, Yun Fa H U. A Density-Based Method for Reducing the Amount of Training Data in kNN Text Classification[J]. Journal of Computer Research & Development, 2004, 41(4):539-545.
4. Yu Pingping, Ni Jiancheng, Yao Binxiu, et al. Efficient KNN Chinese Text Classification Algorithm Based on Spark Framework [J]. Journal of Computer Applications, 2016, 36 (12): 3292-3297.
5. Cheng Kefei, Zhang Cong. Naive Bayes Classifier Based on Feature Weighting[J].Computer Simulation, 2006, 23 (10): 92-94.
6. Li Ronglu, Wang Jianhui, Chen Xiaoyun, et al. Chinese Text Classification Using Maximum Entropy Model[J]. Journal of Computer Research and Development, 2005, 42 (1): 94-101.
7. Yuan R, Li Z, Guan X, et al. An SVM-based machine learning method for accurate internet traffic classification[J]. Information Systems Frontiers, 2010, 12(2):149-156.
8. Ding Zhenguo, Li Jing, Zhang Zhuo. An Improved Text Classification Algorithm Based on Neural Network[J]. Application Research of Computers, 2008,25(6): 1639-1641.
9. Wang Yu, Wang Zhengou. Text Classification Rule Extraction Based on Fuzzy Decision Tree[J]. Journal of Computer Applications, 2005, 25(7): 1634-1637.
10. Liu Wenjun, Zheng Guoyi, Zhang Xiaoqiong. Sample Classification Algorithm Based on Rough Set and Statistical Learning Theory[J]. Fuzzy Systems and Mathematics, 2015, 29(1): 183-190.
11. Li Tao, Liu Bin. Research on Efficient Web Text Classification System under Spark Platform[J]. Computer Applications and Software, 2016, 33 (11): 33-36.
12. Luo Yuanshuai. Research on parallel text classification algorithm based on random forest and Spark[D]. Southwest Jiaotong University, 2016.
13. Sahami M, Dumais S, Heckerman D, et al. A Bayesian Approach to Filtering Junk E-Mail[J]. Papers from the Workshop Aaai, 1998.
14. Su Sui, Lin Hongfei, Ye Zheng. Spam Filtering Based on Character Language Model [C]// National Conference on Information Retrieval and Content Security. 2008:41-47.
15. Pan Wenfeng. Content-Based Spam Filtering Research [D]. Institute of Computing Technology, Chinese Academy of Sciences, 2004.
16. Wang Xiujun,Shen Hong. An Efficient Text Classification Algorithm Based on Incremental Learning Vector Quantization[J]. Chinese Journal of Computers, 2007, 30(8): 1277-1285.
17. Bratko A, Cormack G V, Lynam T R. Spam Filtering Using Statistical Data Compression Models[J]. Journal of Machine Learning Research, 2006, 7(4):2673-2698.
18. Giannakopoulos G, Palpanas T. Revisiting the effect of history on learning performance: the problem of the demanding lord[J]. Knowledge & Information Systems, 2013, 36(3):653-691.
19. Wang J, Zhao Z Q, Hu X, et al. Online group feature selection[C]// International Joint Conference on Artificial Intelligence. 2014:1757-1763.
20. Dietterich T G, Domingos P, Getoor L, et al. Structured machine learning: the next ten years[J]. Machine Learning, 2008, 73(1):3.
21. Valentini G, Masulli F. Ensembles of Learning Machines.[C]// Italian Workshop on Neural Nets-Revised Papers. Springer-Verlag, 2002:3-22.
22. Karpathy A, Toderici G, Shetty S, et al. Large-Scale Video Classification with Convolutional Neural Networks[C]// IEEE Conference on Computer Vision and Pattern Recognition. IEEE Computer Society, 2014:1725-1732.
23. Gharehchopogh F S, Khaze S R, Maleki I. A New Approach in Bloggers Classification with Hybrid of K-Nearest Neighbor and Artificial Neural Network Algorithms[J]. Indian Journal of Science & Technology, 2015, 8(3):237.
24. Kim Y. Convolutional Neural Networks for Sentence Classification[J]. Eprint Arxiv, 2014.
25. Collier N. Convolutional Neural Network Architectures for Matching Natural Language Senences[J]. 2015, 3:2042-2050.
26. S Lai，L Xu，K Liu .Recurrent Convolutional Neural Networks for Text Classification
27. AlexRudnicky. Can Artificial Neural Networks Learn Language Models?[C]// The Proceedings of the. 2000:202-205.
28. Bengio Y, Vincent P, Janvin C. A neural probabilistic language model[J]. Journal of Machine Learning Research, 2006, 3(6):1137-1155.
29. Song Fuxing. Design and Implementation of Spark-Based Super Large Text Classification Method [D]. Beijing Jiaotong University, 2017.
30. Zhou Qinqiang, Sun Bingda, Wang Yi. Research on Text Preprocessing Method of Text Automatic Classification System[J]. Application Research of Computers, 2005, 22(2): 85-86.
31. Liu Keqiang. Analysis and Use of ICTCLAS of 2009 Shared Edition[J]. Science and Education Journal, 2009(22):271-271.
32. Huang Yibiao. Comparative Study of Chinese Word Segmenter for Implementing Lucene Interface [J]. Science & Technology Information, 2012(12):246-247.
33. He Jinfeng. Research on Text Preprocessing Based on Chinese Information Retrieval[D]. University of Electronic Science and Technology of China, 2008.
34. Sun Dianzhe, Wei Haiping, Chen Yan, Realization and Evaluation of Chinese Word Segmentation of

Dismemberment of Ox by Paodin in Nutch[J]. Computer and Modernization, 2010(6):187-190.

35. Ye Na. Research on Text Preprocessing and Rule Automatic Learning Technology for Information Extraction[D]. Northeastern University, 2005.
36. Zhang Ning. Semantic-Based Chinese Text Preprocessing Research [D]. Xidian University, 2011.
37. Li Ying. Research on Text Preprocessing Method Based on Part of Speech Selection[J]. Intelligence Science, 2009, 27 (5): 717-719.
38. Luo Yuanshuai. Research on Parallel Text Classification Algorithm Based on Random Forest and Spark[D]. Southwest Jiaotong University, 2016.
39. Feng Shuxiao, Xu Xin, Yang Chunmei. New Progress of Chinese Word Segmentation Technology in China[J]. Journal of Intelligence, 2002, 21 (11): 29-30.
40. Long Shuquan, Zhao Zhengwen, Tang Hua. Overview of Chinese Word Segmentation Algorithm[J]. Computer Knowledge and Technology, 2009, 5(4): 2605-2607.
41. Zhai Fengwen, He Fengling, Zuo Wanli. Chinese word segmentation method combining dictionary and statistics[J]. Journal of Chinese Computer Systems, 2006, 27(9): 1766-1771.
42. Liu Yang, Guo Qiaojin, Zhou Pengfei, et al. A Spark-based rapid extraction method for massive text keyword: CN 106202556 A [P].2016.
43. Xu Yang, Liu Gongshen, Meng Kui. Text Vectorization Algorithm Based on the Relationship between Words in Sentences [J].Information Security and Communication Privacy, 2014 (4): 84-88.
44. Xiong Dakang. Research and Implementation of Chinese Short Text Classification Technology [D]. Anhui University, 2014.
45. Zheng Lin, Xu Dehua. Text Classification Research Based on Improved TFIDF Algorithm[J]. Computer and Modernization, 2014 (9): 6-9.
46. Zhang Yufang, Peng Shiming, Lu Jia. Improvement and Application Based on Text Classification TFIDF Method[J]. Computer Engineering, 2006, 32 (19): 76-78.
47. Zhang Jinrui. Research and Application of Text Classification Based on LDA[D]. Zhengzhou University, 2016.
48. Guo Lantian, Li Yang, Mu Dejun, et al. A Topic Discovery Method Based on LDA Theme Model[J]. Journal of Northwest Polytechnic University, 2016, 34 (4): 698-702.
49. Wei Qiangshen. Domain Keyword Extraction: Combining LDA with Word2Vec[D]. Guizhou Normal University, 2016.
50. Feng Guichuan. Text Modeling Based on Word2vec and Classification Research[D]. Shenzhen University, 2016.
51. Sui Hao. Research on the Recognition and Tendency Judgment of Weibo Emotional New Words Based on Word2Vec[D]. Guangxi University, 2016.
52. Yan Jiaming. Research and Application of Text Classification Based on Cloud Computing[D]. Zhejiang Sci-Tech University, 2016.
53. Ren Yitian. Research on Massive Text Classification Parallelization Technology Based on Support Vector Machine[D]. Beijing Institute of Technology, 2016.