Wang et al. BMC Systems Biology 2018, 12(Suppl 6):106
https://doi.org/10.1186/s12918-018-0625-3

BMC Systems Biology

RESEARCH Open Access

Recognition of bacteria named entity using conditional random fields in Spark

Xiaoyan Wang[1], Yichuan Li[1], Tingting He[1], Xingpeng Jiang[1*] and Xiaohua Hu[1,2*]

From IEEE International Conference on Bioinformatics and Biomedicine 2017
Kansas City, MO, USA. 13-16 November 2017

Abstract

Background: Microbe plays a crucial role in the functional mechanism of an ecosystem. Identification of the interactions among microbes is an important step towards understand the structure and function of microbial communities, as well as of the impact of microbes on human health and disease. Despite the importance of it, there is not a gold-standard dataset of microbial interactions currently. Traditional approaches such as growth and co-culture analysis need to be performed in the laboratory, which are time-consuming and costly. By providing predicted candidate interactions to experimental verification, computational methods are able to alleviate this problem. Mining microbial interactions from mass medical texts is one type of computational methods. Identification of the named entity of bacteria and related entities from the text is the basis for microbial relation extraction. In the previous work, a system of bacteria named entities recognition based on the dictionary and conditional random field was proposed. However, it is inefficient when dealing with large-scale text.

Results: We implemented bacteria named entity recognition on Spark platform and designed experiments for comparison to verify the correctness and validity of the proposed system. The experimental results show that it can achieve higher F-Measure on the comparison of correctness. Moreover, the predicting speed is much faster than the previous version in large-scale biomedical datasets, and the computational efficiency is improved remarkably by about 3.1 to 6.7 times.

Conclusions: The system for bacteria named entity recognition solves the inefficiency of the previous proposed system on large-scale datasets. The proposed system has good performance in accuracy and scalability.

Keywords: Spark, Named entity recognition, Text mining, Microbial interactions

Background

Microbes are almost everywhere in the global environment. Soils, plant, water and animals are the environment of one or more microbial communities. A variety of microbial communities formed by the aggregation of different proportions microorganisms are commonly referred to as the microbiome. Microbes in the microbiome frequently interact with other members of the community, and these interactions reflect the overall structure and function of the microbial community [1]. Microbes are closely related to host health. Unbalance in microbial communities will lead to a variety of diseases. For example, the microbiome affects the host by making it susceptible to central nervous system autoimmune diseases [2]. Studying the relationships between microbes and diseases provides a new potential to cure a number of diseases. For instance, gastrointestinal microflora can affect fat storage, and thus recovering gut microflora to a healthy state which is helpful for solving the obesity-related problems [3]. In the past 10 years or so, researchers have developed a variety of computational methods for mining a large number of microbial interactions from metagenome abundance data. For example, using the Fisher's exact test to infer whether species co-occur or co-exclusion from spatial metagenomic survey data [4], using the Spearman, Pearson and other correlation coefficients to identify the correlation

* Correspondence: xpjiang@mail.ccnu.edu.cn; huxiaohua@mail.ccnu.edu.cn
[1]School of Computer, Central China Normal University, Wuhan, Hubei, China
Full list of author information is available at the end of the article

between microbial species, or using the LSA algorithm to infer directional interactions from temporal metagenomic data [5]. On the other hand, a large number of microbial interactions validated by many biological experiments are reported in mass biomedical literature and which are often overlooked. Mining these interactions and collating them into a database will create a valuable resource for current research. As one of the main ways to show results and exchange academic results, biomedical literatures accumulate rapidly and its scale is far exceeding those of other disciplines. In particular, there are over 2 million articles related to bacteria studies. How to effectively use these massive data to quickly and accurately discover valuable information are becoming an important part of current research. There are still few studies on how to find out the interactions between microbes from mass biomedical literature. Freilich et al. [6] studied the interactions between microbes based on the co-occurrence of species in the text and constructed an approximate model of the bacterial ecosystem. Lim et al. [7] used support vector machine(SVM) to classify and determine whether there is positive or negative interaction between the given microbial species, which greatly reduces the manual annotation workload, but cannot determine the mode or direction of interactions.

One of the basic tasks of text mining is named entity recognition, which aims to automatically identify the proper nouns. The identification of microbial named entities remains a challenging task, due to the lack of standard corpus, the emergence of new named entities, the existence of phenomena that one entity with different writings and long entities nesting short entities. Named entity recognition (NER) approaches mainly include rule-based methods, dictionary-based methods, and machine learning-based methods. The current mainstream method for NER is machine learning, and of them conditional random field (CRF) is an excellent algorithm among them. In our previous work [8], we manually annotated datasets and proposed a bacteria named entity recognition system with good performance based on the dictionary and CRF. However, for the massive biomedical literature that needs to be identified, the system will encounter a series of challenges in big data processing, including huge computational time and space requirements.

Transferring large-scale computing tasks to the distributed cluster platform has become an effective way to solve the above problems. Spark is a memory-based parallel framework, which will cache the data that will be used repeatedly to the memory to reduce the data loading time. In addition, for the given task, Spark will build a Directed Acyclic Graph (DAG) which tightly arranges calculations and calculations. Hence the framework is able to automatically optimizes tasks according to the logical relationship between operators. The same iterative machine learning algorithm runs faster in Spark than Hadoop by 10~ 100 times. [9]. Therefore, the execution efficiency of the Spark framework is relatively superior. Literature [10] proposes a parallel ant colony optimization (ACO) algorithm based on Spark for combinatorial optimization in the era of big data, which is more than 10 times faster than that based on MapReduce. Literature [11] achieves parallelized frequent item sets mining algorithm based on Spark, and compared it with the algorithm implemented based on MapReduce on a number of benchmark experiments. The experimental results show that the former has an average speed of 18 times faster than the later.

Based on the previous results [8], we proposed a parallel bacteria named entity recognition system based on Spark platform and CRF. The experiment shows that the speed of the Spark version has been greatly improved, with higher time efficiency and good scalability. This lays a foundation for the extraction of bacteria interactions from medical literature.

Materials and methods

Experimental environment and data sets

The experimental environment is as follows: Debian, 3.16.0–4-amd64, Intel(R) Xeon(R) CPU E5–2670 v3 @ 2.30GHz processors, 256GB RAM, Apache Spark 2.2.1, Scala-2.11.8 and JDK1.8.0_71. We built a Spark application with a Stand-alone cluster task scheduling mode on a 48-core server. The CRF algorithm used in the experiments is an open source CRF algorithm in Spark [12]. They use Adam and AdaGrad optimizer based on Spark, so it will get better performance compared with other methods [13, 14].

The datasets used are the corpus (IOB2 format) that are manually annotated in our previous work [8] for bacteria named entity recognition and the 50,000 unannotated biomedical abstracts downloaded on PubMed with the keyword “human”, “oral”, “bacteria”.

Methods

In this paper, we mainly study the computing platform for bacteria named entity recognition based on the conditional random field and Spark. To begin with, we extracted 34 features such as word features, affix features, etc. We trained the CRF model on a training sets in Spark, and then evaluated the model’s performance on a test set. Finally, we compared the Spark version and CRF++ on single node under the same conditions to verify the efficiency of the system, and tried to apply them to large-scale unannotated corpus to compare the prediction speed of them.

Spark computing framework

Representative batch systems include MapReduce [15], Spark [9], Pregel [16] and Trinity [17], etc. Among them, Spark is implemented in Scala language and compatible with Hadoop's original ecosystem while overcoming the shortcomings of MapReduce in iterative computing and interactive data analysis. In addition, it has the advantages of scalability, high reliability and load balancing, and has a huge community support, so it has become the most active and efficient general computing platform for large data. Resilient Distributed Dataset (RDD) [18] is the core data structure of Spark, the scheduling order of Spark is formed by the dependency of RDD, and entire Spark program is formed by the operation of RDD. With such memory calculation mode, Spark supports machine learning and other iterative computing well and has better computational efficiency than MapReduce.

Conditional random field

The conditional random field was first proposed by Lafferty et al. in 2001 [19], which is a discriminant undirected graph model that models the conditional probabilities according to the given observation sequence of variables. In the field of biomedicine, linear chain CRFs are generally used to process sequence labeling tasks such as named entity recognition and part-of-speech tagging and so on.

Assuming X and Y are random variables, P(Y| X) is the conditional probability distribution of Y given X. If the random variable Y constitutes a Markov random field represented by an undirected graph G = (V,E),

$$P(Y_v|X, Y_w, w\neq v) = P(Y_v|X, Y_w, w \sim v) \quad (1)$$

that is, Eq. (1) holds for any node v, then the conditional probability distribution P(Y|X) is called a conditional random field.

In Eq. (1), w~v denotes all nodes w that have edges connected to node v in the graph G = (V, E), w ≠ v represents all nodes other than the node v, and Y_V、Y_u、Y_w are random variables corresponding to node v、u、w.

Assume that $X = (X_1, X_2, ..., X_n)$and $Y = (Y_1, Y_2, ..., Y_n)$ are all random variable sequences represented by linear chains. If given a random variable sequence X, the conditional probability distribution P(Y| X) of the random variable sequence Y constitute a conditional random field, which means Markov Property is satisfied:

$$P\left(Y_i|X, Y_1, ..., Y_{i-1}, Y_{i+1}, ..., Y_n\right) = P\left(Y_i|X, Y_{i-1,}, Y_{i+1}\right) \quad (2)$$

where i = 1, 2, ..., n (Only one side is considered when i = 1 and n).

Then P(Y| X) is a linear chain conditional random field. In the labeling problem, X represents the input observation sequence, Y represents the corresponding output sequence or state sequence. Under the condition that random variable X is x, Y is y, the parametric form of the conditional probability is as follows:

$$P(y|x) = \frac{1}{Z(x)} \exp\left\{\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right\} \quad (3)$$

$$Z(x) = \sum_{y} \exp\left\{\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} u_l s_l(y_i, x, i)\right\} \quad (4)$$

Where t_k and s_l are eigenfunctions, their value is 1 when the feature is satisfied, 0 otherwise. λ_k and $u_l$are the corresponding weights. Z(x)is a normalization factor, summation is done on all possible output sequences. The conditional random field is completely determined by the eigenfunction and corresponding weights. The main tasks of training are feature selection and parameter estimation. The purpose of feature selection is to choose a feature set that can express this random process, and the parameter estimation is to estimate the weights for each feature selected. The training process can be essentially attributed to the process of estimating the weight parameters of the eigenfunctions based on the principle of maximum likelihood function. When the model training is completed, the maximum likelihood distribution and model parameters are obtained. For the new observation sequence X, the most likely output sequence Y is predicted based on training model. The conditional random fields can make full use of contextual label information to achieve good labeling results.

The computational scale of the conditional random field in training is related to the size of training set, templates and the number of output tags. The sequence of input sentences in biological texts is generally very long, so there exists the problems of long time excution of optimization and large memory occupation when training on large-scale data. Research on the efficiency of CRF in handling massive data has become one of the most popular hotspots in biomedical named entity recognition. Literature [20] implements CRFs training on large-scale parallel processing systems based on multi-core and can process large data sets with hundreds of thousands of sequences and millions of features, which significantly reduces the computation time. At the same time, using a second-order Markov-dependent in the training process, the model has achieved higher accuracy; Literature [21] deals with complex computing tasks by decomposing the learning process into smaller and simpler sub-problems. It developed a core approach to learn CRF structure and parameters and speeded up the regression by using more and more parallel platforms. Literature [22] controls the

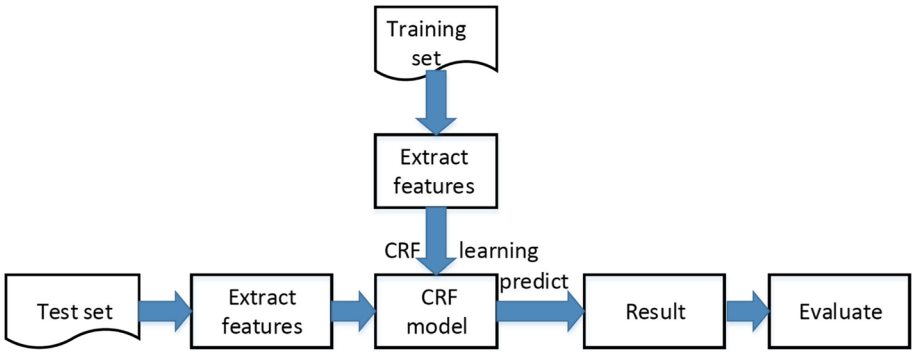

Fig. 1 The Bacteria named entity recognition system flow chart

number of non-zero coefficients by introducing penalties in the CRFs model. Ignoring execution time, it implements CRF's training task on processing hundreds of output tags and up to several billion features; In literature [23], CRF-RNN, a new neural network is proposed based on mean-field approximation and Gaussian potential functions for CRFs. And they obtained the best result of the challenging Pascal VOC 2012 segmentation benchmark when applying the proposed method to the semantic image segmentation problem. Literature [24] achieves the MapReduce-based parallel training of CRFs and can ensure the correctness of the training results. Meanwhile, it greatly reduces the training time and improves the performance. Although this MapReduce-based implementation can handle large-scale training sets and feature sets, the execution efficiency is not high enough. Literature [25] converts all data into RDDs and stores them in the memory of the cluster nodes. It implements SparkCRF, a distributed CRFs running in a cluster environment. Experiments show that SparkCRF has high computing performance and good expansibility, and it has the same accuracy level as the traditional single-node CRF++.

Design and implementation of the system

The proposed system is written in Scala. Firstly, we extracted the features from the data sets on the Spark platform. The features used are the optimal 34 sub-features selected by the single optimal combination method in our previous work [8], and a feature matrix was generated in the next step. The training and predicting steps were executed using the Open Source Toolkit of CRF based on Spark(We call it "Spark-CRF"). The flow chart of the bacteria named entity recognition system is shown in Fig. 1.

The system includes two stages in the workflow: training and prediction. Spark-CRF creates RDDs in nodes and the user-defined Transformation and Action are used for preprocessing, feature extraction, model training and prediction.

Evaluation metrics

Precision (P), Recall (R) and F-Measure (F) are generally used to evaluate the performance of NER system. They are defined as follows, respectively.

Table 1 The performance of models trained on different scale training sets

Training set (The number of sentences)	CRF++ on single node			Spark version		
	Precision	Recall	F-Measure	Precision	Recall	F-Measure
1000	84.679%	73.429%	78.654%	86.715%	80.566%	83.527%
2000	85.442%	76.391%	80.664%	88.031%	80.880%	84.304%
3000	86.287%	78.232%	82.062%	88.623%	81.463%	84.892%
4000	85.707%	78.591%	81.995%	88.389%	82.002%	85.076%
5000	86.447%	78.725%	82.405%	88.699%	81.373%	84.878%
6000	87.831%	80.341%	83.919%	89.492%	82.944%	86.094%
7000	88.456%	80.476%	84.277%	89.981%	83.438%	86.586%
8000	87.745%	80.341%	83.880%	90.398%	83.662%	86.900%
9000	88.345%	80.969%	84.496%	90.847%	84.201%	87.398%
10,000	88.873%	81.373%	84.958%	90.944%	83.842%	87.249%

Table 2 The average prediction time of CRF++ on single node vs Spark version

Data sets (The number of abstracts)	(s)	Spark version (different numbers of processor cores) (s)			
		12	24	36	48
2000	362.411	118.479	83.758	75.223	72.375
10,000	1716.569	533.486	325.471	286.723	268.614
20,000	3081.027	964.063	612.743	525.29	517.477
30,000	5207.298	1406.216	883.148	793.282	734.974
40,000	6141.149	1858.607	1168.061	1020.059	966.032
50,000	7956.735	2154.872	1465.193	1243.926	1191.362

$$\mathrm{P} = \frac{TP}{TP + FP} \tag{5}$$

$$\mathrm{R} = \frac{TP}{TP + FN} \tag{6}$$

$$F = \frac{2 \times P \times R}{P + R} \tag{7}$$

Here, TP is the number of bacteria named entities that are correctly identified by the model, FP is the number of bacteria named entities which are incorrectly identified by the model, FN is the number of non-bacteria named entities that are incorrectly identified by the model. P represents the precision, R represents the recall rate, and F-Measure is the average of P and R.

Results and discussion

This article mainly carried out the following two experiments:

In order to verify the classification performance of the proposed Spark version, we choose to compare the proposed system to CRF++ on single node [8] in terms of the P, R and F-Measure on the same datasets. Taking the first 1000, 2000, 3000, ..., and 10,000 sentences of the manual annotated training set [8] to form 10 training sets for model training. The Spark version performs better than the previous results (Table 1). We can also see that with the increasing scale of the training data, the F-Measure increases for both systems on the whole.

We investigated the effectiveness and scalability of the Spark version by adjusting the scale of application datasets and the number of processor cores. We randomly selected 2000 abstracts, 10,000 abstracts, 20,000 abstracts, 30,000 abstracts, 40,000 abstracts, and 50,000 abstracts respectively in the unannotated texts to form 6 datasets. The number of processor cores is gradually increased from 12 to 48 each time. Each experiment was conducted 5 times repeatedly and the average execution time was recorded.

Table 2 demonstrates that with the increasing scale of the datasets, the average prediction time of both the CRF++ on single node and Spark version is increased accordingly. While the former has many difficulties in dealing with a large amount of data. For different datasets, the prediction time curves of the Stand-alone version and the Spark version (with a 48-cores processor) are shown in Fig. 2. From which we are able to find out that the Spark version runs faster than the CRF++ on single node on the same dataset. With the increasing scale of the datasets, the difference of execution time between the two systems is getting larger and larger and the speed enhancing performance of the Spark version increased significantly. Comparing the prediction time of the stand-alone version and Spark version on the unannotated datasets, it turns out that the speed of the Spark version has been increased by about 3.1 to 6.7 times.

The relationship between the prediction time and the number of processor cores on 6 datasets is shown in Fig. 3, which shows that the larger the dataset, the longer the running time under the same number of processor cores; the larger the number of processor cores, the lesser the execution time under the same dataset. This indicates that our proposed Spark version has good scalability.

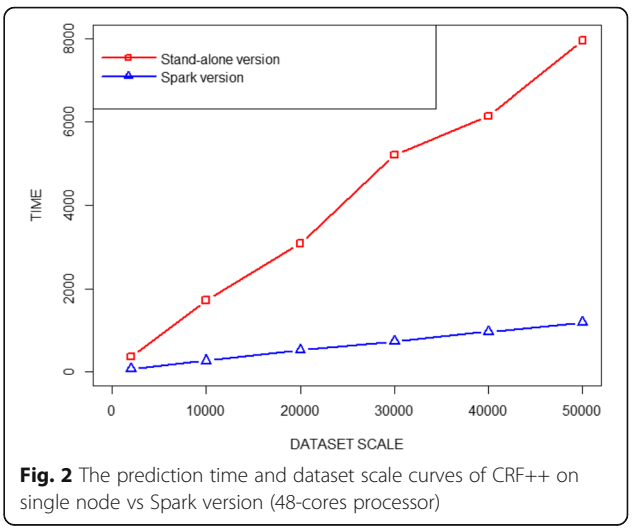

Fig. 2 The prediction time and dataset scale curves of CRF++ on single node vs Spark version (48-cores processor)

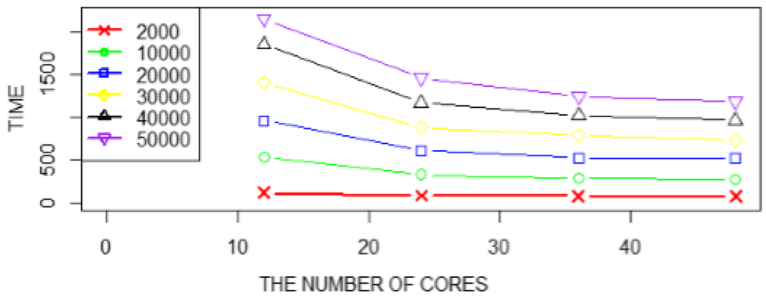

Fig. 3 The prediction time and the number of processor cores curves on 6 data sets

Conclusions

This paper provides a computational system of bacteria named entity recognition based on the dictionary and conditional random fields on the Spark platform. The system includes the procedure of text preprocessing, feature extraction, model training and prediction. We also designed experiments to verify the classification accuracy and time efficiency. Under the large-scale dataset, the proposed system is more effective than the previous Stand-alone version (CRF++ on single node). And its efficiency can be further improved with the expansion of cluster computing ability, which shows good scalability. The training sets and test sets used are limited in scale, however, we haven't verified whether datasets with larger scales would lead to the decrease of accuracy.

Acknowledgements
X.J. thank Mengwen Liu for helpful discussions.

Funding
The research was supported by the National Key Research and Development Program of China (2017YFC0909502), the National Natural Science Foundation of China (61532008, 61872157), and the Self-determined Research Funds of CCNU from the Colleges' Basic Research and Operation of MOE (No. CCNU16KFY04).

Availability of data and materials
Our datasets mainly consist of unannotated corpus and manually annotated corpus: 50000 unannotated abstracts is retrieved on PubMed with "human", "oral", and "bacteria" as the key words, which are mainly used to compare the prediction time of CRF++ on single node and Spark version on large corpus; 1344 annotated abstracts that was manually annotated in our previous work [8] is used for model training and evaluation (https://github.com/bluelilywxy/BacNER-V1.0.git).

About this supplement
This article has been published as part of BMC Systems Biology Volume 12 Supplement 6, 2018: Selected articles from the IEEE BIBM International Conference on Bioinformatics & Biomedicine (BIBM) 2017: systems biology. The full contents of the supplement are available online at https://bmcsystbiol.biomedcentral.com/articles/supplements/volume-12-supplement-6.

Authors' contributions
XJ and XW designed the bacteria named entity recognition under Spark big data platform. XW and YL implemented the system and designed experiment to comparasion and analysis results. XJ and XW contributed to writing the manuscript. TH and XH supervised and helped conceive the study. All authors read and approved the final manuscript.

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests. The publication costs are funded by the National Key Research and Development Program of China (2017YFC0909502).

Publisher's Note

Author details
[1]School of Computer, Central China Normal University, Wuhan, Hubei, China. [2]College of Computing and Informatics, Drexel University, Philadelphia, PA, USA.

Published: 22 November 2018

References

1. Li C, Lim KMK, Chng KR, Nagarajan N. Predicting microbial interactions through computational approaches. Methods. 2016;102:12–9.
2. Wang Y, Kasper LH. The role of microbiome in central nervous system disorders. Brain Behavior Immunity. 2014;38(5):1.
3. Ley RE, Cohen M. Obesity and the human microbiome. Curr Opin Gastroenterol. 2010;26(1):5.
4. Chaffron S, Rehrauer H, Pernthaler J, Von MC. A global network of coexisting microbes from environmental and whole-genome sequence data. Genome Res. 2010;20(7):947–59.
5. Ruan Q, Dutta D, Schwalbach MS, Steele JA, Fuhrman JA, Sun F. Local similarity analysis reveals unique associations among marine bacterioplankton species and environmental factors. Bioinformatics. 2006; 22(20):2532–8.
6. Shiri F, Anat K, Isacc M, Uri G, Roded S, Eytan R. The large-scale organization of the bacterial network of ecological co-occurrence interactions. Nucleic Acids Res. 2010;38(12):3857–68.
7. Lim KMK, Li C, Chng KR, Nagarajan N. @MInter: automated text-mining of microbial interactions. Bioinformatics. 2016;32(19):2981.
8. Wang X, Jiang X, Liu M, He T, Hu X. Bacterial named entity recognition based on dictionary and conditional random field. IEEE Int Conf Bioinform Biomed. 2017:439–44.
9. Zaharia M, Chowdhury NMM, Franklin M, Shenker S, Stoica I, Zaharia M, Chowdhury M, Franklin MJ, Shenker S, Stoica I: SAP, VMware, and Yahoo!. Spark: Cluster Computing with Working Sets. 2010.
10. Zhaoyuan W, Hongjie W, Huanlai X, Tianrui L. Ant colony optimization algorithm based on spark. J Comp Applic. 2015.
11. Qiu H, Gu R, Yuan C, Huang Y. YAFIM: a parallel frequent Itemset mining algorithm with spark. In: Parallel and Distributed Processing Symposium Workshops; 2014. p. 1664–71.

12. Hqzizania M, Vinceshieh, Chenghao-Intel, Ynxiang imllib-spark [DB/OL] 2017. https://github.com/Intel-bigdata/imllib-spark.
13. Kingma DP, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
14. Duchi J, Hazan E, Singer Y. Adaptive subgradient methods for online learning and stochastic optimization. J Mach Learn Res. 2011;12(7):257–69.
15. Dean J, Ghemawat S. MapReduce: simplified data processing on large clusters. ACM. 2008.
16. Malewicz G, Austern MH, Bik AJC, Dehnert JC, Horn I, Leiser N, Czajkowski G. Pregel: a system for large-scale graph processing. Abstract. 2010;18(18):135–46.
17. Shao B, Wang H, Li Y. Trinity: a distributed graph engine on a memory cloud. In: ACM SIGMOD International Conference on Management of Data; 2013. p. 505–16.
18. Zaharia M, Chowdhury M, Das T, Dave A, Ma J, Mccauley M, Franklin MJ, Shenker S, Stoica I. Resilient distributed datasets: a fault-tolerant abstraction for in-memory cluster computing. In: Usenix conference on networked systems design and implementation; 2012. p. 2–2.
19. Lafferty JD, Mccallum A, Pereira FCN. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Eighteenth International Conference on Machine Learning; 2001. p. 282–9.
20. Phan HX, Nguyen ML, Horiguchi S, Inoguchi Y, Ho BT: Parallel training of CRFs: a practical approach to build large-scale prediction models for sequence data. 2013.
21. Bradley JK. Learning large-scale conditional random fields (Doctoral dissertation): Carnegie Mellon University; 2013. Retrieved from http://reports-archive.adm.cs.cmu.edu/anon/ml2013/CMU-ML-13-100.pdf.
22. Lavergne T, Cappé O, Yvon F: Practical Very Large Scale CRFs 2010:504–513.
23. Zheng S, Jayasumana S, Romeraparedes B, Vineet V, Su Z, Du D, Huang C, Torr PHS. Conditional random fields as recurrent neural networks, Proceedings of the IEEE international conference on computer vision. 2015. p 1529-1537.
24. Tao L, Lin L, Luo C. A Parallel Training Research of Chinese part-of-speech tagging CRF model based on MapReduce. Acta Sci Nat Univ Pekin. 2013; 49(1):147–52.
25. Zhu J, Jia Y, Xu J, Qiao J, Wang Y, Cheng X. SparkCRF: a parallel implementation of CRFs algorithm with spark. J Comp Res Dev. 2016;53(8): 1819–28.