A Parallel Computing Method for Entity Recognition based on MapReduce

Yushui GENG[1, a], Peng LI[2, b], Jing ZHAO[3, c]

[1]School of Information, Qilu University of Technology, Jinan, 250353, China

[2]School of Information, Qilu University of Technology, Jinan, 250353, China

[3]School of Information, Qilu University of Technology, Jinan, 250353, China

[a]gys@qlu.edu.cn, [b]brunomarslee@yeah.net,[c]zj@qlu.edu.cn

Keywords: Entity Recognition; Parallel Computing; MapReduce; Hadoop

Abstract. With the rapid development of industrial automation, there are huge amounts of duplicate data refer to the same entity in the data sets have brought enormous challenges in data analysis. To accommodate the entity recognition of huge amounts of data, this paper presents a parallel computing method for entity recognition based on MapReduce. Through the detailed introduction to the MapReduce framework, running the applications on the Hadoop platform and parallel processing data sets to recognize the data entities. The experiments show that the proposed method greatly enhanced the recognition speed and accuracy, which has better effectiveness to meet the demand for entity recognition than other methods.

1. Introduction

With the coming of the era of big data, many types of large quantities of data have been produced in the industrial field. Structured, unstructured and semi-structured data are constantly increasing at an unprecedented rate, which brings great difficulties to make better use of the data for the enterprise. In order to effectively deal with these data resources, it is necessary to make a certain degree of data fusion or mining. However, due to the information update data quickly and a wide variety of data sources, it makes the data accumulate and can not be updated in time, and then appears outdated phenomenon. At the same time, the heterogeneity of different data sources makes the data exist quality problem.

Because of the difference of locating of object and information, the information types from different data sources are various, and the descriptions of the same entities are not the same. And the purpose of entity recognition is to identify the tuples of the same real world entity from the data sets. Entity recognition results can be widely used in other stages of data quality management.

2. Related Work

Entity recognition is mainly spent to determine whether two data record descriptions for the same entity object in the real world. Early entity recognition algorithm is mainly to detect duplicate records in order to obtain the recognition results. At present, the algorithm is mainly based on the similarity function[1] and rule[2]. Literature [3] takes advantage of user annotations example to study the transformation rules of the string, which can improve the precision; puts forward a kind of language called Deduplog, which is a life style, unrelated areas and own the ability to define entity recognition rules. Literature [4] is based on the MapReduce framework of the geometric similarity research, [5] mainly summarizes the data block technology. Based on the real data sets, the paper [6] evaluates the efficiency of entity unity. In addition, there are other aspects of the research, like heuristic method [7], distance function [8], Markoff chain [9] and so on. At present, the latest entity recognition method using machine learning [10] algorithm, machine learning library has brought the realization of many classifiers.

Entity recognition technology is mainly through some calculation rules or laws, using a calculation method to identify the possibility of an entity, whether it is the same one before. As

entity recognition is of great significance in data quality management, the research of entity recognition has also been paid enough attention to. However, despite the existing methods can effectively identify the entity in many applications, but there are still many deficiencies: (a) currently, entity recognition exists duplicate names and synonyms; (b) traditional entity recognition method is often based on tuple similarity comparison to obtain the results; (c) at present, entity recognition method in the used of similarity measure does not take into account the correlation between different words. Based on the background of the era of big data, Hadoop is currently more popular effective tool for processing large data. And HDFS and MapReduce provide an effective data storage and efficient data processing mode for big data solutions.In order to solve the problem of entity recognition, this paper presents a parallel processing method for entity recognition based on MapReduce.

3. Parallel Processing Method for Entity Recognition

3.1 MapReduce Framework

The idea of MapReduce programming is to break down large tasks into small tasks which will then be performed respectively, so as to achieve the purpose of reducing processing time. Using the parallel framework MapReduce on Hadoop to conduct a preliminary screening for input data sets, so that to remove the less likely record pairs of the input data sets in a relatively short period of time. The parallel computation saves a lot of work time and improves the efficiency for entity recognition.

Split operation is based on the case of the source file and divided into a series of InputSplit in accordance with the specific rules, each InputSplit will be processed by a Mapper. Splitting the files is not to split the file to form new file fragmentation copies, but to form of a series of InputSplit. InputSplit contains the data information, such as the file block information, the starting position, the data length, the node list, etc. Therefore, all the data of the splits will be find just according to InputSplit. The most important task of Split operation is to determine the parameter SplitSize, SplitSize is the size of the split data. Once it is determined, the source file is divided in turn according to the value. If the file is less than this value, then the file will become a separate InputSplit; if the file is larger than this value, then it would be divided in accordance with the SplitSize, left less than SplitSize become part of a separate InputSplit. The rule for determining the SplitSize value is: SplitSize= max{minSize, min{maxSize, blockSize}}. Thus, the size of SplitSize is between SplitSize and blockSize.

Mapper receives the <key, value> form of data and processes into <key, value> form. The specific process can be defined by the user.

From the results of Mapper direct output to it becomes to the final Reducer direct input data after a series of processing, the above process is the whole process of Shuffle and it is also the core process of MapReduce. The whole process of Shuffle can be divided into two stages: the Mapper-side Shuffle and the Reducer-side Shuffle. The data generated by the Mapper is not directly written to disk, but first to storage in memory; when memory data reaches the set threshold, then the data is written to the local disk and simultaneously do the sort, combine, partition, etc. Sort operation is to sort the results that generated by Mapper according to the key value; combine operation is to combine the adjacent records with the same key value; partition operation involves how to evenly distribute the data to multiple Reducers, it is directly related to the load balance of Reducer. Among them, the combine operation does not necessarily to have, because it is not applicable in certain scenarios; but in order to make the output results of Mapper more compact, it will be used in most cases.

Reducer accepts data stream in the form of <key, {value, list}>, makes the data in form of <key, value> and output them, then the output data will be written directly to the HDFS. The specific process can be defined by the user.

3.2 The Process of Entity Recognition

The method of this paper is mainly by introducing weight and similarity and using Hadoop platform and MapReduce framework, which will process data into the form of key-value data pairs

and can be efficiently applied to the entity recognition. The flow chart as shown in Figure 1.

Step 1: Pretreatment process

Taking the objects described by data as the entity, and preprocessing the data in the data sets. K fields of each data are selected as the key and the entire data record as the value to constitute form of <key, value>. The k fields can be selected as shown in Table 1.

Table 1: Four fields of each data

No.	Name	ID	Price	Color
0842	sugar	S4426	null	white
9426	baking soda	BS270	0.37	white
1735	null	S4426	0.65	white
0058	salt	S712	0.13	white
3710	soda	S245	0.46	white
0714	sugar	S4426	0.65	white

Step 2: Forming data pairs

Computing the Cartesian product of each data sets, that is, each data and any other one will be made pairs respectively, and to constitute the form of data pairs. For example, there are four data like L, M, N, O, and then the six data pairs will be composed of LM, LN, LO, MN, MO, NO.

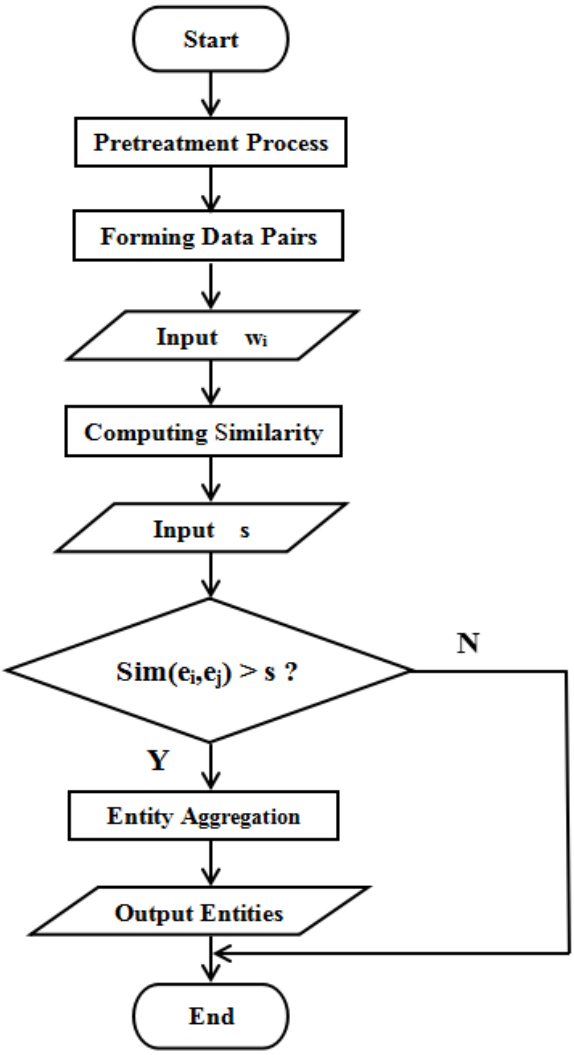

Fig.1. The flow chart of entity recognition

Step 3: Computing the similarity

For entity e_i and e_j, the more similar the content of the k fields information are, the more close to the same entity the two entities will be. The k fields are given corresponding weight w, and the w of

each field is different. For the entity e_i, the more the decisive factor of the field is, the greater the value of its w_i will be. Then, according to the importance of the field, the weight of each part can be set as shown in Table 2.

Table 2: Weights of each data

	Name	ID	Price	Color
weight	0.49	0.63	0.16	0.12

The similarity of each entity pair can be calculated according to the values of w_i, its equation is:

$$Sim(e_i, e_j) = \frac{\sum (w_i, w_j)}{\sum w_i}, (0 < i < n; 0 < j < n; w_i > 0; w_j > 0) \quad (1)$$

Computing the similarity value and then screening each data pairs. Set the similarity threshold is s, only the entity which has reached the specified threshold will enter step 4, and all the other data will end this process.

Step 4: Entity aggregation

All the data pairs that reached the threshold will be unified, that is to merge the same entity pairs into one data. This process is to merge the same kind data and form a data set with unified entities. The results can be shown in Table 3.

Table 3: The recognition results

No.	Name	ID	Price	Color
0824	sugar	S4426	null	white
1735	null	S4426	0.65	white
0714	sugar	S4426	0.65	white

After the above four steps, the entities are gradually recognized from the vast amounts of data. And the end results will only have the entities that have been recognized.

4. Experiment and Analysis

In this paper, the experiment environment is based on Hadoop operating platform, related algorithms and programs running on the Hadoop-2.7.2. Experiment builds Hadoop cluster under the Linux environment, configuring the main node and slave nodes, and starting the Hadoop cluster successfully. The cluster contains one namenode and four datanodes, each node has 512MB memory and 1GB hard disk. Using CentOS6.5 system, virtual machine software VMware, virtual machine RedHat5.3, while using Xmanager tools to facilitate the operation. Experiment data from a manufacturing enterprise with 13,174 items and a electronic commerce website with 8,392 items. In order to evaluate the accuracy of the method, comparing the recognition results of this paper with entity recognition method based on rules, using Precision(P), Recall(R) and F1-measure as the measurement standard. And P=A/(A+B), R=A/(A+C), F1-Measure=(2PR)/(P+R). The five kinds of data sets of book, movie, staff, commodity and material information are selected as entities to carry out the experiment. The experiment chooses the method based on rules and the method based on probability as the comparison objects to test the effect of three methods respectively at different data size in the parallel cluster environment. The results are shown in Figure 2 and Figure 3.

The experiment has obvious effect in the following two aspects: (1) Under the conditions of the same datanodes, the parallel computing method of this paper has a great advantage compared to the other. According to the red line in Figure 2, advantages of the method based on MapReduce are more obvious. (2) Under the conditions of the same entity object, the accuracy of the method of this paper is relatively high, and it has a high degree of recognition to the entity. The red line in Figure 3 shows that the effect of the parallel computing method is better than the other.

According to the above analysis of the graphs and the results, it is apparent to draw a conclusion that the method based on MapReduce has greatly accelerated the speed of entity recognition and also improved the efficiency, so that it has sufficient advantage to be used to real life. Thus it is concluded that Hadoop platform operating environment and MapReduce Framework can be well

used for entity recognition.

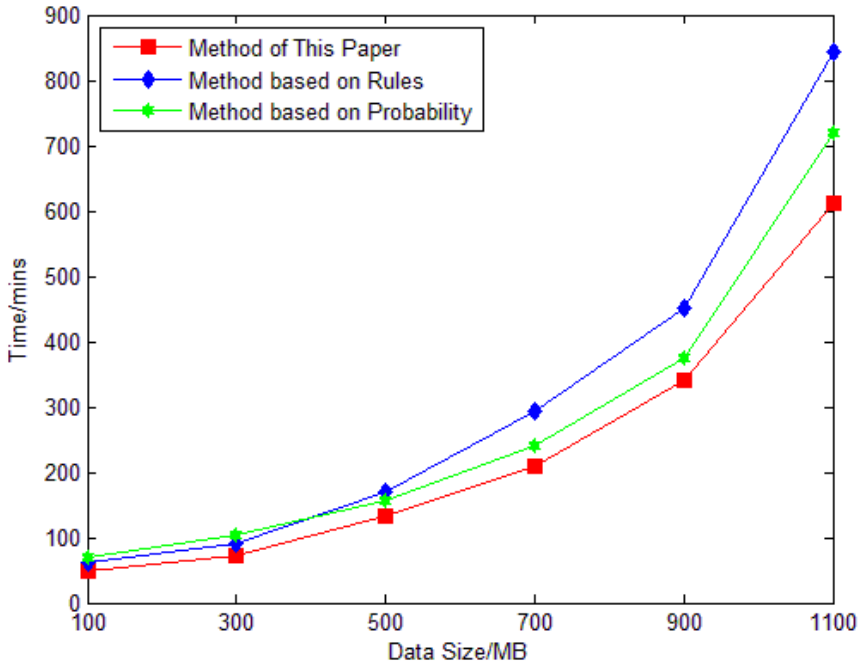

Fig.2. The speed results

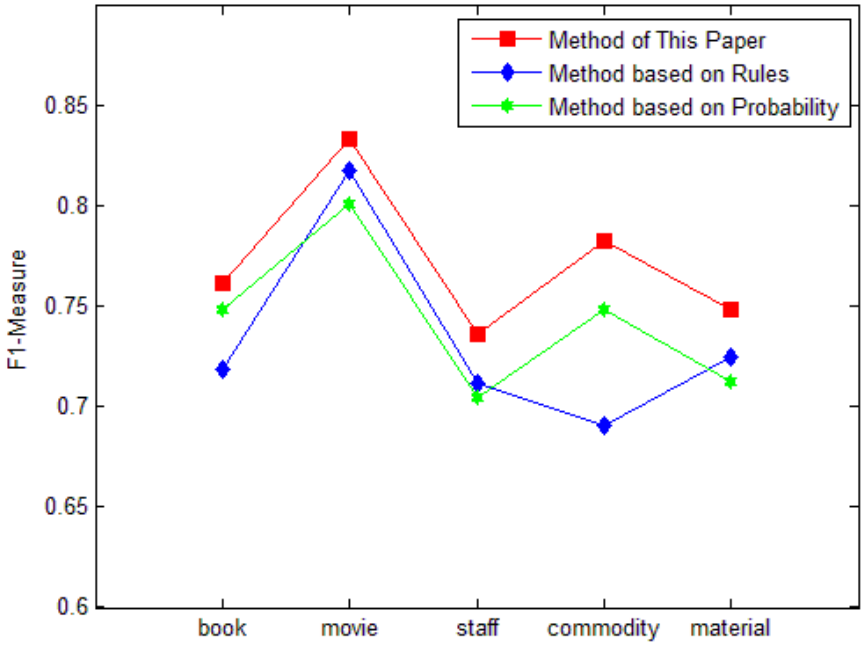

Fig.3. The accuracy results

5. Summary

This paper proposes a parallel computing method for entity recognition, whose realization relies on introducing weight, similarity and using Hadoop platform and MapReduce framework. Experiments show that the proposed method greatly improves the speed and accuracy of entity recognition. However, due to the limitations of the data sources and other constraints, there are still a lot of work need further in-depth research and more attention will be paid to other algorithms and computational models so as to better applied to entity recognition.

6. Acknowledgments

[1]Science and Technology Development Plan Project of Shandong Province (No.2014GGX101052)

[2]Independent Innovation and Achievements Transformation Special Project of Shandong Province (No.2014ZZCX03408)

[3]Shandong Provincial Natural Science Foundation, China (No.ZR2014FQ021)

[4]Key Research And Development Plan Project Of Shandong Province (No.2015GGX106003)

[5]A Project of Shandong Province Higher Educational Science and Technology Program (No.J15LN03)

7. References

[1] Arasu A, Chaudhuri S, Kaushik R. Learning string transformations form examples. Proceeding of the VLDB Endowment,2009,2(1):514-525

[2] Koudas N, Saha A, Srivastava D, et al. Metric functional dependencies.IEEE International Conference on Data Engineering,2009:1275-1278

[3] Arasu A, Re C, Suciu D, Large-Scale Deduplication with Constraints Using Dedupalog[J].IEEE International Conference on Data Engineering,2009:952-963

[4] Vernica R, Garey MJ, Li C, Efficient parallel set-similarity joins using MapReduce[J].Acm Sigmod International Conference on Management of Data,2010:495-506

[5] Kirsten T, Kolb L, Hartung M, et al. Data Partitioning for Parallel Entity Matching[J]. Proceedings of the VLDB Endowment,2010,3(2):1-12

[6] Pcke H, Thor A, Rahm E. Evaluation of entity resolution approaches on real-world match problems[J].Proceedings of the Vldb Endowment,2010, 3(1-2):484-493

[7] Arasu A, Chaudhuri S, Kaushik R. Learning String Transformations From Examples[J]. Proceedings of the Vldb Endowment, 2009, 2(1):514-525

[8] Chen Z,Kalashnikov DV,Mehrotra S.Adaptive graphical approach to entity resolution[C]. Acm/ieee Joint Conference on Digital Libraries, 2007:204-213

[9] Singla P, Domingos P.Entity Resolution with Markov Logic[C].Proc of IEEE ICDM'06. piscataway,NJ:IEEE,2006:572-582

[10] Ghoting A,Krishnamurthy R, Pednault E, et al.SystemML: Declarative machine learning on MapReduce.IEEE International Conference on Data Engineering, 2011, 6791(4):231-242