A Method of Large - Scale Log Pattern Mining

Lu Li[1(✉)], Yi Man[1], and Mo Chen[2]

[1] School of Electronic Engineering, Beijing University of Posts and Telecommunications,
Beijing 100876, China
SallyLi0863@163.com

[2] Institute of Network Technology, Beijing University of Posts and Telecommunications,
Beijing 100876, China

Abstract. With the development of the telecommunication network, more and more devices are used in the network, which has been a burden for the network operation and maintenance. At the same time, network devices generate large amounts of log data every day, recording the activities of each device in detail. As a result, the log can reflect the performance of network state, and sometimes, we can predict the occurrence of network failure based on the log. However, since the log has such features: big volume, multi-source heterogeneous and difficult to understand, people have not reasonably used it to analyze and predict network failure. Therefore, we propose a method for structuring a large number of device logs in the short term, and use the data generated from a real communication device network to verify the effect. Besides, we compare our method with the traditional log parsers, such as regular expressions, LogSig, etc. to demonstrate the efficient processing performance and accurate pattern extraction analysis for massive network device logs.

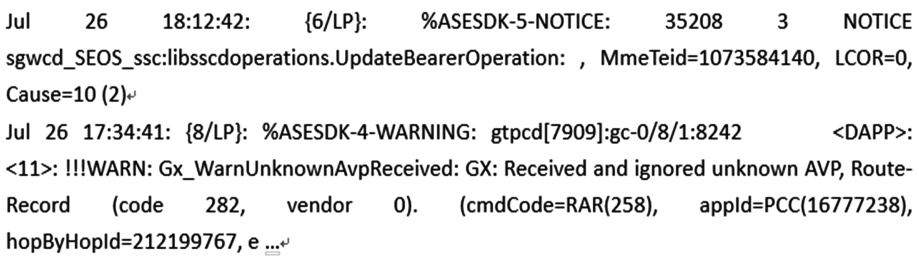
Keywords: Big data · Log parser · Telecommunication network equipment
Word2vec

1 Introduction

With the rapid development of telecommunications technology, the telecommunications network is more complex and the network business is more diverse. At the same time, the mining for a large amount data generated by network has also attracted the attention of many people. Network devices log contains a lot of information, which can represent the operating state and healthy degree of network. However, because of the volume and characteristic of the log data, the researchers have not achieved remarkable results. For example, all the equipment in an operator can produce about 2 TB log data in a province in one day, and these log are written by seven different vendors with different formats (Fig. 1).

Obviously, without the instructions and the guidance of the professionals, raw log message produced by telecommunication network equipment, as shown in the following example, is difficult for the operator to understand the exact meaning of these logs, not to mention using it to carry out further work.

Q. Zu and B. Hu (Eds.): HCC 2017, LNCS 10745, pp. 76–84, 2018.
https://doi.org/10.1007/978-3-319-74521-3_9



```
Jul 26 18:12:42: {6/LP}: %ASESDK-5-NOTICE: 35208 3 NOTICE
sgwcd_SEOS_ssc:libsscdoperations.UpdateBearerOperation: , MmeTeid=1073584140, LCOR=0,
Cause=10 (2)↵
Jul 26 17:34:41: {8/LP}: %ASESDK-4-WARNING: gtpcd[7909]:gc-0/8/1:8242 <DAPP>:
<11>: !!!WARN: Gx_WarnUnknownAvpReceived: GX: Received and ignored unknown AVP, Route-
Record (code 282, vendor 0). (cmdCode=RAR(258), appId=PCC(16777238),
hopByHopId=212199767, e ...↵
```

Fig. 1. Typical raw telecommunication device log data

At present, there are several methods to deal with log, and methods based on the pattern are widely used in log analysis. In this method, one raw log message can be divided into two parts: the constant part and the variable part. For telecommunication network equipment log, the variable part contains a lot of valid information, such as the location of the module that issued the log, the actions performed by the operator, and the time of the log. However, when the volume of log increases to a certain extent, due to the huge variable data, despite using this method, the results will take up a lot of storage space.

Therefore, we investigated the various log preprocess methods and for the characteristics of the network device logs, we have taken certain ways based on the natural language model, making complex log becomes more suitable for storing and mining. In order to validate our method, we used it and several other typical log parser to compare the result in the test set of different kinds of logs and the real network device data set, which proves the accuracy and efficiency of our method.

2 Log Parser

2.1 Parser Methods

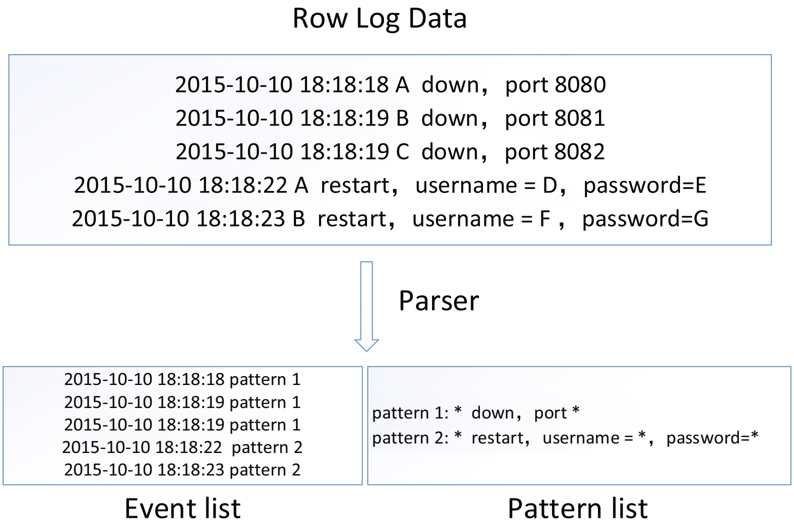
There are three kinds of methods that are mainly used for log data parser.

2.1.1 Methods Based on Regular Expression

In the traditional log processing methods, regular expression is often used to extract a specific field. Many programming languages support the use of regular expressions for string manipulation. It can develop the structured data to process the log, so that a large number of non-structural or semi-structured information is discarded. And this kind of method is not flexible enough, basically for some specific log need to be processed.

2.1.2 Methods Based on Pattern

The log data is automatically generated by the program in the device, which is often composed by constant strings and variable parameters, so the log data has obvious semi-structured features. By generating and comparing the existing set of patterns, the words in the log are divided into log template words and variable words, so that we can find the abnormal parameters in the data set (Fig. 2).



```
2015-10-10 18:18:18 A  down,  port 8080
2015-10-10 18:18:19 B  down,  port 8081
2015-10-10 18:18:19 C  down,  port 8082
2015-10-10 18:18:22 A  restart,  username = D,  password=E
2015-10-10 18:18:23 B  restart,  username = F ,  password=G
```

```
2015-10-10 18:18:18 pattern 1
2015-10-10 18:18:19 pattern 1
2015-10-10 18:18:19 pattern 1
2015-10-10 18:18:22  pattern 2
2015-10-10 18:18:23 pattern 2
```

```
pattern 1: *  down,  port *
pattern 2: *  restart,  username = *,  password=*
```

Fig. 2. The object of methods based on pattern

For example, in [5], the author proposes a STE method to judge whether it is a log template word or a variable word based on the location and length of the word in the log text, which can determine the log pattern.

This is the usual method for log analysis, and its speed is faster and performance is better, but because the device log parameters varies and the data format is very irregular, leading to the poor quality and large redundancy for the pattern set, which has a great impact on the effect of the data mining work.

2.1.3 Methods Based on Data Mining

At present, many studies apply the data mining algorithm to the log process, for example, [6] applied the k-center clustering algorithm to analyze the ITS system log data to analyze the internal structure of the ITS system. In [7], by using the clustering algorithm, a log pattern recognition algorithm based on distributed platform is designed to improve the speed and efficiency of log recognition.

However, at present there is not suitable data mining method for telecommunication network device log analysis.

2.2 Three Typical Parsers

In order to verify the performance of our parser, we chose three typical parsers to compare with ours. And these parser's source code can be find in the [1].

2.2.1 LKE

This method is proposed to parse free-form text log for anomaly detection in distributed systems, and it is made up by the following steps: (1) remove the parameters according to the established rules (2) measure raw log similarity by the weighted edit distance (3) cluster similar raw log keys together (4) Log template generation [2].

2.2.2 IPLoM

This method is proposed for automatic event log analysis, which includes three step hierarchical partitioning process: (1) Partition by log length. The first step is to use the token count heuristic to partition the log messages, because the log messages that have the same line format are likely to have the same token length (2) Partition by token position. By counting the words in the same position, the method will sort the log by the count. (3) Partition by search for mapping. By searching for mapping relationships between the set of unique tokens in two token positions, the log is divided into smaller partition. (4) Log template generation [3].

2.2.3 LogSig

To understand and optimize system behaviors, LogSig is proposed to generate system events from textual log messages. LogSig works in three steps: (1) Word pair generation. Raw log data are converted to a set of word pairs to record both the word and its position. (2) Log Clustering. Based on the word pairs, a potential value is calculated for each log message to decide which cluster the log message potentially belongs to. (3) Log template generation. In each cluster, the log messages are leveraged to generate a log template [1, 4].

2.3 Our Parser

Telecommunication device log data often have the following characteristics: (1) Complex parameters. Most of the parameters are numbers. (2) Short text. Most of logs are short sentences or parameters list, and the sentence are irregular. The longest sentence is no more than 30 words. (3) Difficult to understand. Without professional instruction book, it is difficult to identify the meaning of the log.

According to the characteristics of the raw data, our analytical method has three steps. First, remove all the punctuations, numbers and the words containing numbers. Second, compute the Hash value of the processed text, to obtain unique Hash value of each log text. Third, by comparing the hash values, the log is merged into the same log pattern, and we can obtain the log pattern table and the simplified log event sequence. Fourth, use the edit distance to merge the log patterns again and rewrite the log event sequence.

2.4 Parser Practice

2.4.1 In the Five Kinds of Log Data Set

We use five different log data, which come from different log systems (BGL, HPC, HDFS, Zookeeper and Proxifier), each kind log data contains two thousand lines [1], and we used our parser and three other parsers to compare the result. In the experiment, we used the same environment and code language to come to the following results.

From the results we can see that for the small data set of log data, IPLoM algorithm show a great advantage in speed, our algorithm is running faster than the other two parsers. In terms of accuracy, our parser has an advantage in the analysis of certain logs (Figs. 3 and 4).

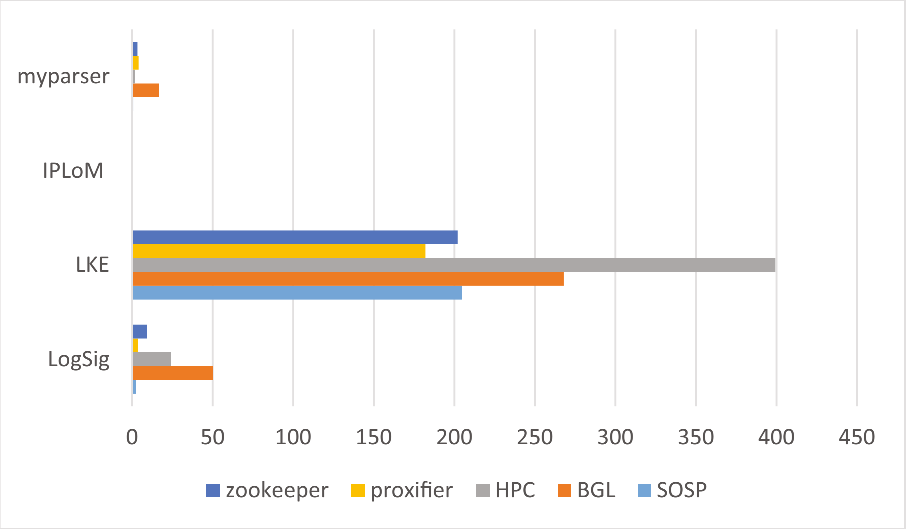


Fig. 3. The running time of four parsers

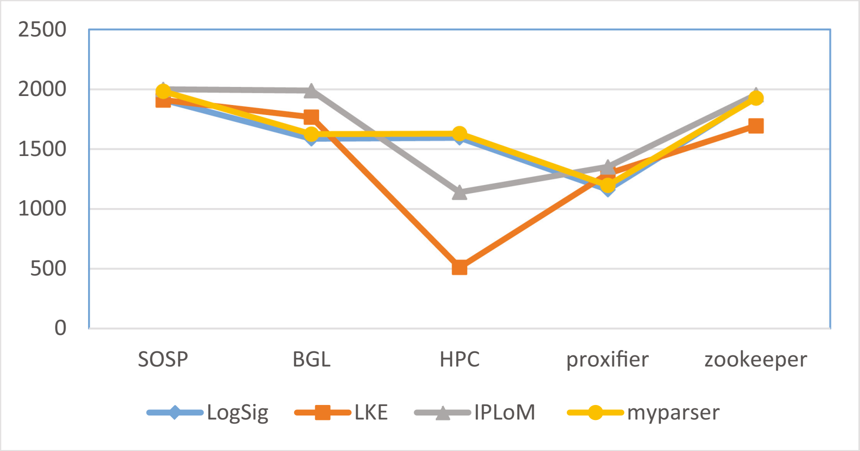


Fig. 4. The accuracy of four parsers

2.4.2 In the Large-Scale Log Data Set

We still experimented with actual data set on, because the actual data set is very large, we use thousandth log in one day, 750 M data, containing more than 7 million log data (Table 1).

Table 1. The running time of four parsers on big data set

Parser name	Time(s)
LogSig	38581
LKE	–
IPLoM	376
My parser	7673

We listed the running time of each parser, where LKE was unable to perform log parsing due to memory overflow, and we could see that IPLoM still had a great advantage in speed, but when there was no standard regular expressions, its results contain a lot of redundancy. Our parser will give less redundant and more accurate results within a tolerable time range.

2.4.3 Summary

Comparing our performance with the other three classic parsers, we found that although our parser was not as fast as IPLoM, it showed good adaptability and speed when dealing with a large number of telecommunications device logs data. After that, we have carried out further data mining to understand the data and find the information in the log data.

3 Log Analysis

3.1 Background

3.1.1 Word Vector

To apply the machine learning method to the natural language processing field, the most basic problem is the representation of the language symbol. So far, the most commonly used method for natural language processing is One-hot Representation, which means that n words are n-dimensional vectors, each vector is 1 in a dimension and the other dimension is zero. However, this method will cause the lexical gap problem, there is no connection between words and words.

Therefore, the Distributed Representation method is proposed, that is, using low-dimensional real vector to represent vocabulary. The biggest advantage of Distributed Representation is that it can make meaning-related words relatively similar in distance. At the same time, the word vector will show many special properties, as shown below (Fig. 5).

$$\mathrm{V(King)} - \mathrm{V(Queen)} \approx \mathrm{V(man)} - \mathrm{V(woman)}$$

Fig. 5. An application of word vector

3.1.2 Neural Network Language Model

Bengio Yoshua proposed a neural network algorithm using a three-layer neural network to build the model, the purpose of the model is to predict the next word wt by former n – 1 words. At the bottom are the former n – 1 words (Wt – n + 1 ~ Wt – 1), and C(w) means the vector of the word W. The input layer of the network is to concatenate the n-1 vectors to a (n – 1) * m dimensions vector, which is labeled x. The second layer of the network is obtained directly by using d + Hx, where d is the offset term, and the initialization value is random, using tanh as the activation function (Fig. 6).

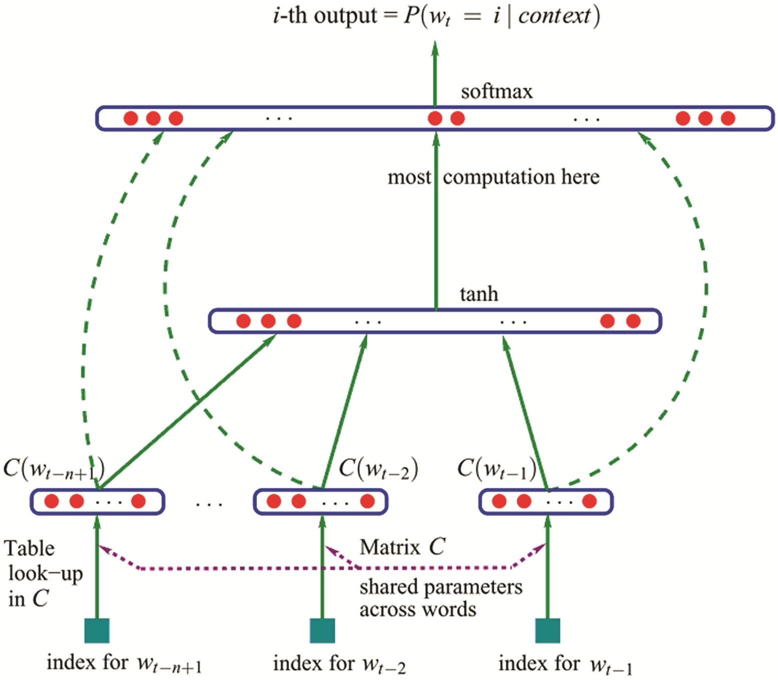


Fig. 6. Neural network structure proposed by Bengio Yoshua

The third layer of the network is represented by the node Yi, using the softmax function to normalize the output value into probability, and the final Y is calculated as:

$$Y = b + Wx + U * \tanh(d + Hx)$$

U is the parameter from the hidden layer to the output layer, the majority of the model compute operation is centered on the matrix multiplication of the U and hidden layers. Finally, we use the stochastic gradient descent method to optimize the model, then we can get word vector [8].

3.1.3 Word2vec

Word2vec is a tool launched by Google to calculate words vector, which has been gained comprehensive attention because of its efficiency and convenience. It is based on the neural network and the natural language model. By the relationship between words and sentences, it can calculate a word vector for each word, and we can compare the word similarity by the distance between the word vectors. Based on the principle of word2vec, we have present a log mining method that can get literally similar logs or logs containing a log of important links.

3.2 Use Word2vec in the Log Process

3.2.1 The Method

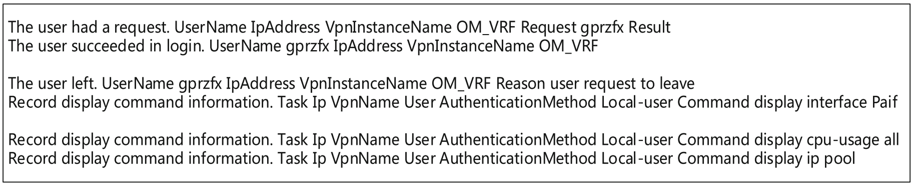
We use our parser to parse certain carrier's seven-day telecommunication device log data. (1) Firstly, we parse the log data into a log pattern set and a log event data table. (2) Then, we list one-day log pattern number in order of their time sequence, as the word2vec sentence, and we see each log pattern as word in word2vec. Through the word2vec tool we calculate the word vector for each pattern. (3) Finally, we derive a similar pattern set by comparing the Euclidean distance between the word vectors and comparing the similarity between patterns.

3.2.2 The Result

We optimized the parameters of word2vec for structure log data. When using the Skip-Gram model, the vector dimension is 50 dimensions and the window size is 5, we find that the word vector is more accurate when looking for similar data patterns. At the same time, we choose the vectors whose cosine similarity is greater than 0.9 as a similar pattern. Finally, we get a number of similar patterns, and these similar patterns is of great significance in the problem analysis (Table 2 and Fig. 7).

Table 2. The parameters of word2vec

CBOW	Size	Window	Negative	HS	Sample	Threads	Binary	Iter
0	50	5	0	1	1.00E−04	20	0	100



```
The user had a request. UserName IpAddress VpnInstanceName OM_VRF Request gprzfx Result
The user succeeded in login. UserName gprzfx IpAddress VpnInstanceName OM_VRF

The user left. UserName gprzfx IpAddress VpnInstanceName OM_VRF Reason user request to leave
Record display command information. Task Ip VpnName User AuthenticationMethod Local-user Command display interface Paif

Record display command information. Task Ip VpnName User AuthenticationMethod Local-user Command display cpu-usage all
Record display command information. Task Ip VpnName User AuthenticationMethod Local-user Command display ip pool
```

Fig. 7. The similar log pattern produced by word2vec

In the results, we can find that some similar log patterns produced by word2vec have literal similarity, which means that word2vec can help us to optimize the log parsing and find else log patterns need to be classified except which has the different digital parameters or whose edit distance is less than a certain value. At the same time, word2vec can help us discover log patterns that are literally unrelated but have similar meanings or links, which is of great importance to subsequent log analysis.

4 Conclusion

Our algorithm uses the neural network language model for the first time to analyze the telecommunication equipment log, and obtains the similarity pattern. At the same time, we design the log analysis method to adapt to the log of the telecommunication equipment, and verify the effectiveness of the method by experiment.

Through the experiment and the comparison of the results, we can find that our parser obtains a better analytical effect for the telecommunications server log data in the tolerable time. Subsequent analysis, whether using word2vec for similar patterns discovery, or the use of other data mining methods to explore, such as abnormal point recognition and correlation analysis, can be based on our processing results for analysis.

However, our analytical methods are also deficient, for example, we can find patterns that have similar characteristics in the order of occurrence, but how these patterns are applied specifically to some telecommunications systems problem, such as system error prediction, auto log analysis system without expert, we also need to continue exploring and researching.

References

1. He, P., et al.: An evaluation study on log parsing and its use in log mining. In: IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE Computer Society, pp. 654–661 (2016)
2. Fu, Q., Lou, J., Wang, Y., Li, J.: Execution anomaly detection in distributed systems through unstructured log analysis. In: Proceedings of International Conference on Data Mining, ICDM 2009 (2009)
3. Makanju, A., Zincir-Heywood, A., Milios, E.: Clustering event logs using iterative partitioning. In: Proceedings of International Conference on Knowledge Discovery and Data Mining, KDD 2009 (2009)
4. Tang, L., Li, T., Perng, C.: LogSig: generating system events from raw textual logs. In: Proceedings of ACM International Conference on Information and Knowledge Management, CIKM 2011, pp. 785–794 (2011)
5. Kimura, T., lshibashi, K., Mori, T., Shiomoto, K.: Spatio-temporal factorization of log data for understanding network events. In: INFOCOM 2014 Proceedings. IEEE (2014)
6. Juneja, P., Kundra, D., Sureka, A.: Anvaya: an algorithm and case-study on improving the goodness of software process models generated by mining event-log data in issue tracking systems. Support. Care Cancer **6**(6), 539–541 (2015)
7. Hamooni, H., Debnath, B., Xu, J., et al.: LogMine: fast pattern recognition for log analytics. In: CIKM (2016)
8. Bengio, Y., et al.: A neural probabilistic language model. J. Mach. Learn. Res. **3**(6), 113 (2003)