An Alarm Correlation Algorithm Based on Similarity Distance and Deep Network

Boxu Zhao(✉) and Guiming Luo

School of Software, Tsinghua University, Beijing, China
boxu.zhao@foxmail.com, gluo@tsinghua.edu.cn

Abstract. Currently, a few alarm correlation algorithms are based on a framework involving frequency and support-confidence. These algorithms often fail to address text data in alarm records and cannot handle high-dimensional data. This paper proposes an algorithm based on the similarity distance and deep networks. The algorithm first translates text data in alarms to real number vectors; second, it reconstructs the input, obtains the alarm features through a deep network system and performs dimension reduction; and finally, it presents the alarm distribution visually and helps the administrator determine the new fault. Experimental results demonstrate that it cannot only mine the correlation among alarms but also determine the new fault quickly by comparing the graphs of the alarm distribution.

Keywords: Alarm correlation · Fault discovery · Word to vector · Similarity distance · Auto-encoder · Deep network

1 Introduction

Alarm correlation is one of coral studies in fault location field, especially in fault location of large complex set of devices, since alarms, resulted in by equipment failures, have strong correlation in different devices or different components of device. As a key technology in telecommunication networks management, it obtains the root cause in a large number of alarms by mining alarm information. A few alarm correlation algorithms have been proposed include Rule-Based Correlation [1], Coding Approach [2], Model-Based Reasoning [3], Case-Based Reasoning [4], Fuzzy Logic [5], Bayesian Networks [6], Neural Networks [7] and so forth. However, those algorithms to mining information often ignore unstructured fields or text data containing important information, also cannot handle the high-dimensional alarm data and fail to provide enough information to help the networker locate and clean the fault.

Some other researchers focused on the data mining methods to analyze alarm sequence and mine alarm correlation rules. Correlation rule mining are also often used to compress the alarms and investigate the root reasons, where frequent episodes of alarms are paid close attention and support-confidence are calculated to obtain correlation rules. Mannila et al. [8] presented the efficient algorithms (WINEPI) for detection of frequent episodes from a given amount of episodes. The algorithms are applied in telecommunication alarm management in TASA [9]. Gardner and Harle [10] studied the generalization of a great deal of network performance information gathered everyday and

D.-S. Huang et al. (Eds.): ICIC 2016, Part III, LNAI 9773, pp. 359–368, 2016.
DOI: 10.1007/978-3-319-42297-8_34

made a tool for network fault discovery by using alarm data in SDH network system. Cuppens and Miege [11] constructed a cooperative module to manage, cluster, merge and correlate alarms for intrusion detection system, significantly reducing the amount of alarms to deal with. Shin and Ryu [12] applied the concept of data mining to alarm correlation, which is helpful not only to supervise the terrible users and hosts but also to find out potential alarm sequences. Xu and Guo [13] presented Alarm Association Algorithms based on Spectral Graph Theory (AAASG), which can reduce the search data set and quickly find fault by the changes in the point structure.

Analyzing more than 5500 sequential alarm records in Management Information System of a Chinese telecom service provider between January 2012 and May 2013, we discover that there are 41 network elements generating 8 different type of alarms. In text data field like “alarm summary”, there are 1401 different values including great amount of information which need efficient data conversion technology to dimensionality reduction. Therefore, the algorithms should be capable of translating the text expression data to digital expression. In alarm correlation, its the key point to discover and position the associated alarms, so clustering algorithms are often made use of so that alarms are divided into different categories and similar property and features of them are in one category. When considering each alarm record as a vector, the “similar” ones are of relatively close distance in metric space. However, number of categories is hard to decide and can varies from one dataset to another, so it is necessary to transform the data in high dimensional space to lower dimensional space and show them visually to help manager realize the correlation relationship quickly and accurately and then locate the fault.

In the context of the paper, we present a new method based on similarity distance and deep network to find correlation of different alarms and compress redundant alarm in telecom network. Initially, text data are separated to words and transformed to high-dimensional vector. Later, we use deep learning method to do unsupervised learning and obtain another optimal expression represent alarm features. Finally, alarm data after processing is visually shown through alarm distribution graph, then administrator can efficiently find out root fault by the changes of graph.

2 Related Work

2.1 Mathematization of Text Data

When considering that machine deals with natural language, mathematical language is necessary and real vector is usually used. A simple method One-hot Representation expresses word by a vector consisting of ‘0’ and ‘1’ as long as the dictionary. While there is only one ‘1’ in the vector, which represents the position of word in dictionary, leading to curse of dimensionality. Furthermore, One-hot fails to take order and relation between words into consideration. In 1986, Hinton [14] put forward ‘Distributed Representation’ concept, that is, every word is mapped to real number vector and distance between vectors represents semantic similarity. When finishing training a language model, word vector is obtained. One method producing word vector is Artificial Neural Network, firstly presented by Xu and Rudnicky [15], Institute of Deep learning, Baidu. Subsequently, a series of related research work has been done including Bengio et al. [16] and Mikolov’s group [17].

2.2 Alarm Distance

After translating word to vector, alarm data can be considered as a high-dimensional Euclidean space $\Omega = \{a_1, a_2, \ldots, a_N\}, a_i \in R^D$. Each alarm is a point in the space of RD. In [13], correlation alarms (association alarms) are defined as follows:

If $\exists A_1, A_2, \ldots, A_i$ are correlation alarms, we say that

$$min\left(\frac{freq(A_1 \bigwedge A_2 \bigwedge \ldots \bigwedge A_i)}{freq(A_1)}, \frac{freq(A_1 \bigwedge A_2 \bigwedge \ldots \bigwedge A_i)}{freq(A_2)}, \ldots, \frac{freq(A_1 \bigwedge A_2 \bigwedge \ldots \bigwedge A_i)}{freq(A_i)}\right) \geq f_{min} \quad (1)$$

Where $freq(A_i)$ is the frequency of alarm A_i occurred in different associated windows. However, this definition ignores two points: (1) A_i is defined as alarm type of a few alarms, or can be seen as an attribute of alarm data, but not the alarm individual. In this paper, we define the alarm individual as a_i, $(a_1, a_2, \ldots, a_N)$ is the sequence of alarms in order of occurrence time. (2) The associated window and alarm frequency only reflect 'time' attribute of alarm data, while there are other attributes like "IP address", "alarm reason", "alarm level" without consideration. Alarm can be seen as a tuple $(t, X_1, X_2, \ldots, X_M)$, t represents occurrence time, $X_1, X_2, \ldots, X_M$ represent other alarm attributes or features, X_{ji} represents the jth attribute of alarm a_i. The alarm similarity distance between a_i and a_j can be defined as:

$$D(a_i, a_j) = \left[\alpha_0 (t_i - t_j)^2 + \sum\nolimits_{k=1}^{M} (X_{ki} - X_{kj})^2\right]^{\frac{1}{2}} \quad (2)$$

$\alpha_0, \alpha_1, \ldots, \alpha_M$ is weight coefficient. When $D(a_i, a_j) > d$, we consider a_i and a_j are correlate.

2.3 Deeping Learning Method

Usually, an alarm record consists of many dimensions, especially after word to vector, a text data dimension is translated into many digital dimensions. That means data may include too much 'unimportant' information which fails to reflect intrinsical features, then it is necessary to acquire 'good' features in some way and try best to ensure the difference between input and output as small as possible at the same time. In Deep Learning, we stack multiple layers and make the output of this layer as the input of the next layer to express the input hierarchically. Through adjusting the parameters in the system, we make the difference between the input and final output as small as possible. In this paper, we use Auto-encoder model capturing the most important factor of input data to find out the main ingredient representing the original information.

3 Method

The algorithm works as follows: (1) Translate text data in alarm record data to real number vector. (2) Combine the result of last step and Non-text data, reproduce the input signals as far as possible and obtain the most important factor representing the original alarm data lossless. (3) Show the result visually. Figure 1 shows algorithm steps.

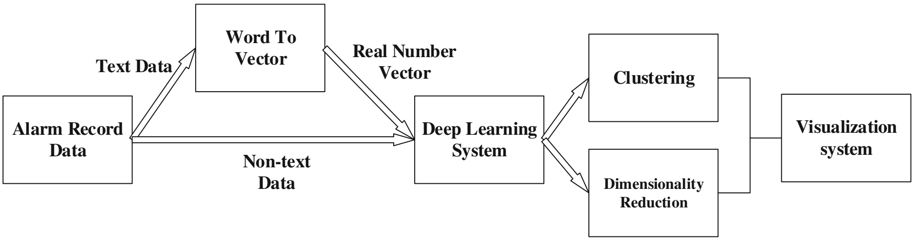

Fig. 1. Algorithm steps

3.1 Text Data to Digital Data

Separate Sentence to Words. Word is the smallest meaningful linguistic component. In Chinese, a sentence consists of several words, a word consists of one or more 'Chinese character' which is basic element of Chinese. Blank space is the natural delimiter in English, but there is no clear delimiter between Chinese words. So Chinese term analysis and parsing is the key of information processing. At present, three main methods have been used for Chinese word segmentation, which include character matching method, statistical method and understanding method [18]. Actual segmentation system usually makes use of character matching method as first step and further improves the accuracy of segmentation through other language information. In character matching, a word is recognized if it is found in a dictionary. Dictionary, text scanning sequence and matching principle are three essential factors. Text scanning order includes forward scanning, reverse scanning and bilateral scanning. Matching principle includes maximum matching, minimum matching, word by word matching and optimal matching. Below is a typical maximum matching algorithm of forward scanning.

Algorithm 1. A typical maximum matching algorithm of forward scanning

```
procedure MMAOFS($S$, $Ws$, $Ml$)
    $S$ represents sentence to be separated, $Ws$ represent output words, $Ml$ represents max length of a word.
    $Ws =$ "";
    while $S$ is not null do
        repeat
            if $Str$ is not in dictionary then
                Remove the character on the Right of $Str$;
            else
                $Ws = Ws + Str+$"\"; $S = S + Str$;
                Break;
            end if
        until $Str$ is a single character
        $Ws = Ws + Str+$"\"; $S = S + Str$;
    end while
    output Ws;
end procedure
```

Translate Words to Vector. After segmentation, next step is translating words into vector so that we can simplify the text processing to vector operations and calculate the similarity in vector space to represent the similarity of text semantics. Word2vec [17] has caused many researchers' attention since it was released free by Google. It uses a three-layer neural network to model the language and get the representation of word in vector space at the same time. Continuous Bag-of-Words Model (CBOW) and Skip-gram Model [19] is used in the language model. The former is a prediction of the current term through known context; on the contrary, the latter is to predict the context through current word. For CBOW and Skip-gram, word2vec gives two framework based on Hierarchical Softmax and Negative Sampling. More details can be seen in [16, 19, 20].

3.2 Autoencoder and Deep Network

An autoencoder is usually a feed-forward neural network aimed to learn a compressed, distributed dataset representation (encoding). The output is trained as a "Representation" of the input, and the input and target data are the same, that is, autoencoder tries to learn a $h(x) \approx x$ function. However, some limits, such as number of hidden layers and hidden neurons, should be added to obtain a meaningful structure. Autoencoder has two advantages: (1) From the output of hidden layers, autoencoder can get some of the compressed representation. For example, if input data has 8 dimensions and one of hidden layers has 4 nodes, 8 dimensional output closed to input should be reconstructed from the 4 dimensional data. (2) Hidden layers' data retains the correlation of the input and makes it easy to observe. Luckily, the features of autoencoder satisfy the need for alarm correlation very well. The compressed data can help show the correlation of alarms visually.

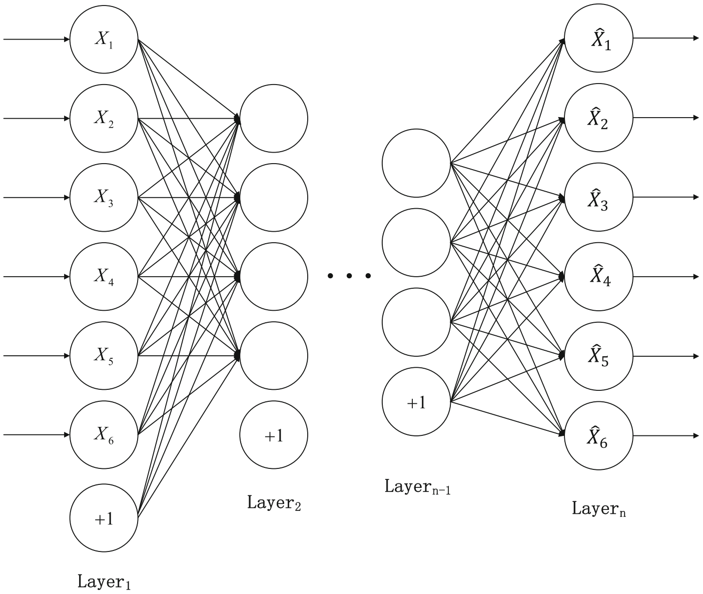

Fig. 2. Deep network of autoencoders

As autoencoder can get the reconstruction of the input data, the hidden layer can be seen as another expression of the original input data. We remove the third layer, and then use the same method to create a three-layer network and make the input of new-constructed network the output of hidden layer. The hidden layer of the second autoencoder is another expression of the input. Following this approach, we create deep neural network composed of a few autoencoders, as shown in Fig. 2.

3.3 Visual System

Alarm Window. Every alarm $(t, X_1, X_2, \ldots, X_M)$ has a time stamp t, *StartTime* < t < *EndTime*, representing alarm occurrence time. Divide the time from *StartTime* to *EndTime* into several segments, each one has the length w. We set 2w the maximum time interval of related events, so that the first alarm and the last alarm are related when we analyses alarms of adjacent time windows. (See Fig. 3) If there are N windows between *StartTime* to *EndTime*, they can be expressed as $w_1, w_2, \ldots, w_M$.

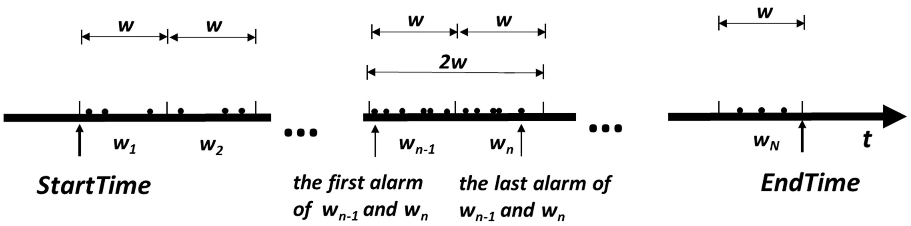

Fig. 3. Alarm window

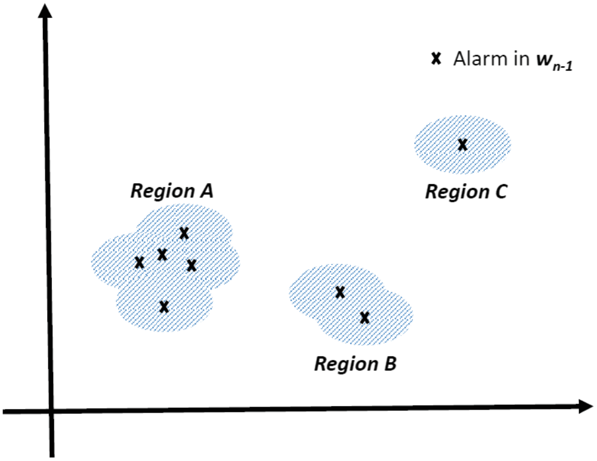

Fig. 4. Three related regions in alarm window w_{n-1}

Related Region. To show alarm records visually, high dimensional alarms in an alarm window are converted into two dimension through deep learning system. Every alarm data point forms a region nearby less than correlation distance d. Some gathering data points form a larger region as they have common area, some scattered points form single region. There are three Related Region below in Fig. 4.

Alarm Attention. When we have alarm distribution graph and related regions during a continuous period, it is easy to classify the alarm data and find new faults. To measure how close we pay attention to each alarm, we introduce the concept Alarm attention. If $a_i \in A_{w_n}$, Alarm attention of a_i can be defined as:

$$Alarm\ attention(a_i) = min\big(D\big(a_i, a_j\big)\big),\ a_j \in A_{w_{n-1}} \tag{3}$$

That means the farther between a_i and related regions in previous time window, the more attention should be paid.

In Fig. 5 we can see No.324, No.325, No.326, No.330 alarm have larger alarm attention than No.327, No.328, No.329, No.331. As we know, alarm distribution has obvious change when a new fault occurs. Therefore, alarms not in related regions have larger attention, that is, the administrator of telecom system should give priority to these alarms and find the fault leading to these alarms.

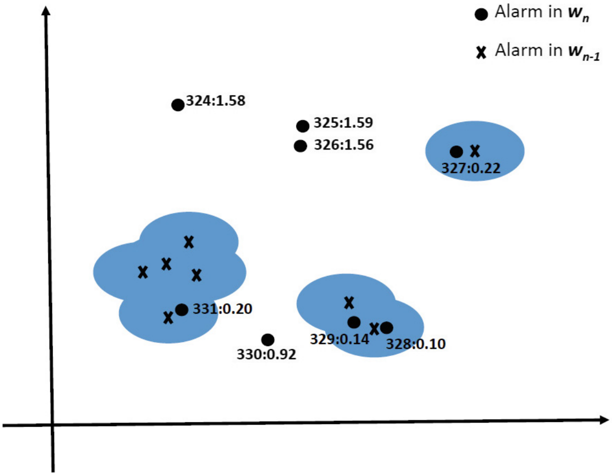

Fig. 5. Alarm in w_n and their alarm attention

4 Experiment

The algorithm is implemented on the dataset in management information system of telecom service providers. Every alarm in the dataset is numbered according to their occurrence order. More than 5500 alarms are produced in 478 days. In the experiment, we set $w = 1\,day$, then there are 478 windows from January 1st, 2012 to April 23rd, 2013.

The key to this algorithm is to compare the alarm distribution between neighbor alarm windows, observe alarm trend and find out system faults leading to these alarms. Alarm identification (IP address of network element, alarm type, alarm level, alarm summary) and the occurrence are selected as alarm attributes. To specifically analyzing two adjacent alarm window we show the alarm point structure of w_{424} and w_{425}. The alarm id and alarm attention is expressed by XXX:XXX in the Fig. 7. The id represents Time sequence of alarm, such that occurs earlier alarm 4855 than alarm 4982.

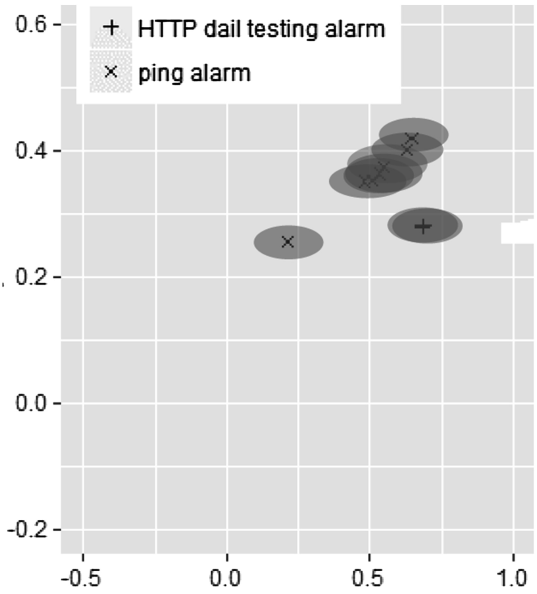

Fig. 6. Alarm point structure and related region in w_{424}

In Fig. 6, different alarms type is expressed by different identifiers. Gray area represents related regions. Overlapping area has deeper color, that means an alarm in next window has more than one correlate alarm when it falls in the area.

Compared to Fig. 6, alarms encircled by dashed line are alarms having larger alarm attention in Fig. 7, that is, the distribution has obviously changed. The change of alarm structure is due to a fault in the system, because the distribution of alarm will be changed when a new fault takes places. When administrator follows alarms of large alarm attention closely, it will be clear how to deal with the new fault.

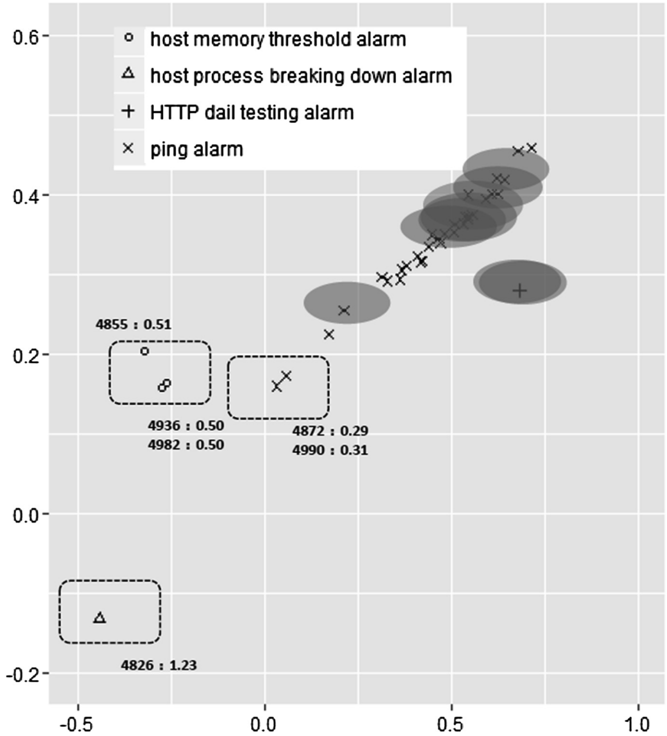

Fig. 7. Alarm point structure and related region in w_{425}

5 Conclusion

This article presents an algorithm based on alarm similarity distance, which is different from the skeleton of frequency and support-confidence. The algorithm focus on each alarm individual, not a type of alarms. Experimental results demonstrate the algorithms has the following advantage: (1) it can deal with the text data of alarm and translate it to vector space, which has not been used before; (2) it can reconstruct the alarm distribution lossless, then get the correlation between alarm accurately according to graph of alarms; (3) it can help administer find out the new fault quickly in system by changes of adjacent windows and clean it without too much effort.

Acknowledgment. This work is supported by the Funds NSFC61572279 and the Science Foundation of Chinese Ministry of Education—China Mobile 2012.

References

1. Cronk, R.N., Callahan, P.H., Bernstein, L.: Rule-based expert systems for network management and operations: an introduction. Network **2**(5), 7–21 (1988)
2. Kliger, S., Yemini, S., Yemini, Y., Ohsie, D., Stolfo, S.J.: A coding approach to event correlation. Integr. Netw. Manage. **95**, 266–277 (1995)

3. Meira, D.M., Nogueira, J.M.S.: Modelling a telecommunication network for fault management applications. In: Network Operations and Management Symposium, NOMS 1998, vol. 3, pp. 723–732. IEEE (1998)
4. Slade, S.: Case-based reasoning: a research paradigm. AI Mag. **12**(1), 42 (1991)
5. Zadeh, L.A.: Fuzzy logic. Computer **4**, 83–93 (1988)
6. Heckerman, D., Mamdani, A., Wellman, M.P.: Real-world applications of Bayesian networks. Commun. ACM **38**(3), 24–26 (1995)
7. Gurer, D.W., Khan, I., Ogier, R., Keffer, R.: An artificial intelligence approach to network fault management. SRI International, 86 (1996)
8. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. Data Min. Knowl. Disc. **1**(3), 259–289 (1997)
9. Hätönen, K., Klemettinen, M., Mannila, H., Ronkainen, P., Toivonen, H.: Knowledge discovery from telecommunication network alarm databases. In: Proceedings of the Twelfth International Conference on Data Engineering, pp. 115–122 (1996)
10. Gardner, R.D., Harle, D.A.: Fault resolution and alarm correlation in high-speed networks using database mining techniques. In: Proceedings of 1997 International Conference on Information, Communications and Signal Processing, ICICS 1997, vol. 3, pp. 1423–1427 (1997)
11. Cuppens, F., Miege, A.: Alert correlation in a cooperative intrusion detection framework. In: 2002 IEEE Symposium on Security and Privacy. Proceedings, pp. 202–215 (2002)
12. Shin, M.S., Ryu, K.H.: Data mining methods for alert correlation analysis. Int. J. Comput. Inf. Sci. (IJCIS) (2003)
13. Xu, Q., Guo, J.: Alarm association algorithms based on spectral graph theory. In: International Joint Conference on Artificial Intelligence, JCAI 2009, pp. 320–323 (2009)
14. Hinton, G.E.: Learning distributed representations of concepts. In: Proceedings of the Eighth Annual Conference of the Cognitive Science Society, vol. 1, p. 12 (1986)
15. Xu, W., Rudnicky, A.I.: Can artificial neural networks learn language models? (2000)
16. Bengio, Y., Schwenk, H., Senécal, J.S., Morin, F., Gauvain, J.L.: Neural probabilistic language models. In: Holmes, D.E., Jain, L.C. (eds.) Innovations in Machine Learning, pp. 137–186. Springer, Heidelberg (2006)
17. Mikolov, T., Le, Q.V., Sutskever, I.: Exploiting similarities among languages for machine translation. arXiv preprint arXiv:1309.4168 (2013)
18. Huang, C., Zhao, H.: Chinese word segmentation: a decade review. J. Chin. Inf. Process. **21** (3), 8–20 (2007)
19. Guthrie, D., Allison, B., Liu, W., Guthrie, L., Wilks, Y.: A closer look at skip-gram modelling. In: Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC-2006), pp. 1–4 (2006)
20. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. J. Mach. Learn. Res. **12**, 2493–2537 (2011)