# Research on Computing Word Similarity in Pre-Qin Classics Language Network Oriented to Digital Humanities

Haotian Hu*, Sanhong Deng** and Dongbo Wang***

* ** *** School of Information Management, Nanjing University, Nanjing 210023, China
Jiangsu Key Laboratory of Data Engineering and Knowledge Service, Nanjing 210023, China
* <hhtdlam@126.com>, ** <sanhong@nju.edu.cn>, ***<db.wang@njau.edu.cn>

Haotian Hu is a PhD candidate at the School of Information Management at Nanjing University, and also at the Jiangsu Key Laboratory of Data Engineering and Knowledge Service. His main research interests include knowledge organization, digital humanities, natural language processing, and informetrics.

Sanhong Deng is a professor at the School of Information Management at Nanjing University, and also at the Jiangsu Key Laboratory of Data Engineering and Knowledge Service. His main research interests include intelligent information processing and retrieval, scientific evaluation, and scientometrics.

Dongbo Wang is a professor at the School of Information Management at Nanjing Agricultural University and also at the Jiangsu Key Laboratory of Data Engineering and Knowledge Service. His main research interests include natural language processing, text mining, infometrics, and digital humanities.

Hu, Haotian, Sanhong Deng, and Dongbo Wang. 2023. "Research on Computing Word Similarity in Pre-Qin Classics Language Network Oriented to Digital Humanities". *Knowledge Organization* 50 (7): 457-474. 29 references. DOI:10.5771/0943-7444-2023-7-457.

**Abstract:** At present, there is relatively little research on ancient Chinese texts in the field of digital humanities, and ancient Chinese information processing urgently needs new algorithms. To realize the word similarity calculation of pre-Qin classics, a total of 25 pre-Qin classics were first mapped into a language network. Based on local relative entropy, we proposed an improved weighted network node similarity calculation method (LREW). This method judges the similarity based on the local network characteristics of the nodes, and the degree of the nodes and the weight information of the edges between the nodes are considered. We used the relative entropy to calculate the difference in the amount of information between different nodes. After experimental comparison, compared with the existing LRE and RE algorithms based on relative entropy, the proposed LREW method can achieve the best results in calculating the similarity between words in the pre-Qin classics. Compared with CN, Jaccard, Salton, and CDSim algorithms based on common neighbor nodes, although the accuracy of LREW is low, the comprehensiveness of the similar word recognition is high, which can ensure that potential similar nodes in the network will not be missed.

Received: 28 November 2020; Revised: 04 August 2023; Accepted 07 September 2023

Keywords: digital humanities; pre-Qin classics; complex network; node similarity; ancient Chinese information processing

## 1.0 Introduction

Computer technology, humanities, and social sciences integrate Digital Humanities. Current main topics of international digital humanities research are natural language processing (NLP), social network analysis (SNA), geographic information systems (GIS) and so on (Chen and Chang 2019). Although computer technology is widely involved in the study of digital humanities, the research on ancient Chinese still lacks tools and urgently needs the innovation of new algorithms (Li et al. 2021).

Word similarity calculation is an intermediate step (Han et al. 2015) and an important field (Guo et al. 2016) in NLP and knowledge mining. It can be viewed as a bridge connecting vocabulary level (Li et al. 2018) and sentence (or chapter) level (Wang et al. 2019) ancient text NLP, helping to realize automatic classification of ancient books, providing semantic-based ancient text retrieval services, etc.

However, there is research on lexical similarity in ancient Chinese. Current research focuses on word similarity computation in Modern Chinese (Yin et al. 2020) or Chinese-English cross-lingual (Wang et al. 2018a) domains. Typically, this requires training word embedding on large-scale labeled datasets and is challenging to transfer to other domains directly.

Pre-Qin classics have great historical and literary value. The knowledge mining of pre-Qin classics can provide new ideas and methods for Ancient Humanities Computing (Huang and Wang 2017). Therefore, based on 25 pre-Qin classics that have been manually segmented and machine-assisted annotated, we explored the design of the complex network node similarity algorithm and its application.

There are three main contributions of our research:

(1) We propose an improved weighted network node similarity calculation method based on local relative entropy (LREW). Compared with previous methods, this algorithm can effectively incorporate edge weights between nodes into the local structural information, highlighting the differences between nodes in the language network.

(2) The LREW method outperforms the relative entropy-based LRE and RE complex network similarity calculation methods. Compared to common neighbor-based methods such as CN, Jaccard, Salton, and CDSim, although LREW scores lower, it can provide a more comprehensive coverage of similar words in the network.

(3) We construct a large-scale Pre-Qin classics Language Network (PQLN) based on 25 pre-Qin classics and comprehensively verify the pros and cons of LREW and baseline methods in calculating similar words. We also discuss in depth the performance differences and potential causes from both macro and micro perspectives.

## 2.0 Related Work

### 2.1 Similarity calculation of Chinese words

At present, the mining of Chinese similar words and synonyms mainly focuses on modern Chinese texts. Guo et al. (2016) used the Word2Vec tool to train word vectors on the Baidubaike corpus, and compared the performance of HowNet-based, Word2Vector-based, etc., methods and the combination of them. Finally, they developed a Chinese character similarity calculation method based on the combination strategy. Huang et al. (2018) integrated prior knowledge into word embedding and combined different similarity computing methods to explore the best combination performance. Ma et al. (2019) proposed a method for identifying similar words in web text. Based on the combination of structural, sentence pattern, and contextual features, an Entity Synonym Network (ESN) was constructed to integrate similarity information. Yin et al. (2020) proposed an enhanced word embedding similarity calculation model EWS-CS, it combines character-level concept, synonym and emotional information into a general pre-training model based on similarity tasks, thus solving the problem of poor similarity word calculation due to lack of domain knowledge.

The mentioned Chinese similar word calculation methods mainly use word embedding technology to represent the contextual information of words. Such methods often require the construction of a large-scale training dataset, and need strong computing resources during vectorization, so it is more challenging to develop. In addition, the word vectors are usually trained on the non-domain corpus, so it is hard for these methods to complete domain-specific similar word mining tasks (e.g. ancient books on agriculture).

### 2.2 Similar words calculation based on network node similarity

Similar words calculation methods based on complex network nodes and network structures have relatively low requirements on corpus size and computer hardware (Wang et al. 2018b), and are not restricted by corpus content (Zhang et al. 2018; Chen et al. 2020). It is one of the important research directions for mining similar words in corpus.

Han et al. (2015) proposed a word similarity calculation method based on contribution discount, which introduced the weight information of the edges between nodes and the global degree feature on the basis of traditional local features. Konaka and Miura (2016) proposed a method based on the Domain Graph where the hypernym of each word is included in the thesaurus and expressed by a directed Domain Graph. The similarity can be obtained by calculating the Jaccard similarity of the two terms' Domain Graph. Hu et al. (2015) proposed a word similarity calculation method based on text mapping and Bayes statistics. By taking the text as a word sequence, it comprehensively examines sentence structure information and vocabulary co-occurrence information. Pablo and Jung (2016) built an etymological graph of Sino-Korean words, and built a Chinese characters graph based on this network. They obtained network structure information through a random forest classifier and distinguish whether character pairs are synonym pairs. Based on graph model, Wang et al. (2015) proposed a similar word search method GBFSM. By mapping the real dataset to WordNet, the Breadth First Search (BFS) method is used to query top-k semantic similar words. Ren and Cheng (2015) designed a heterogeneous graph-based structured entity synonym recognition method.

Under the guidance of the graph-based ranking problem-solving idea, it jointly considers the literal and structural properties of entities to find similar words. In addition, there are some node similarity measurement methods for networks, which can be used for similar word mining in language networks after adjustment. For example, methods based on differences in information entropy (Jiang and Wang 2020; Chen et al. 2020), neighbor nodes (Yang et al. 2017; Zhu et al. 2017), and dynamic time series (Yang et al. 2019).

To the best of our knowledge, there is currently no method based on the node similarity of complex network for similar word mining in pre-Qin classic texts. Therefore, our research on word similarity computing in pre-Qin classics can expand the connotation of digital humanities in ancient texts information processing to a certain extent.

### 2.3 Existing methods of node similarity calculation

Currently, there are two main ideas for calculating the network node similarity. The first is based on common neighbor nodes, with popular methods being CN, Jaccard, Salton, and CDSim. The second idea is based on the difference in information entropy between nodes; the mainstream methods are LRE and RE. In this section, we will introduce the implementation process of these methods respectively and use them as baseline models for experimental comparison.

**(1) CN**

Common Neighbors (CN) is one of the commonly used methods in node similarity calculation (Lü and Zhou 2011). The number of common neighbor nodes between two nodes is the similarity measure. Equation 1 is the calculation process.

Where $a$ and $b$ are two nodes in the network, and $\Gamma(a)$ represents the first-order neighbor node set of $a$. An obvious shortcoming of the CN method is that the large degree node will obtain a higher similarity with the task node than the small degree node, which makes the calculation result biased.

**(2) Jaccard**

The Jaccard method (Güneş at al. 2016) considers the common neighbor nodes while adding local network structure information. Equation 2 is the calculation process of this method.

Based on the CN method, the Jaccard method also considers the influence of the neighbor node of the task nodes. It avoids the problem of false high similarity of large degree nodes to a certain extent, and makes the calculation results more reasonable.

**(3) Salton**

The principle of the Salton method (Li et al. 2018) is similar to that of the Jaccard method. The specific calculation process (Equation 3) divides the result of the CN method by the square root of the modular multiplication of the neighbor nodes of the two task nodes.

**(4) CDSim**

Han et al. (2015) proposed a word similarity calculation method based on contribution discount (CDSim), incorporating the weight between node pairs. The calculation of contribution $ctrb(a,p)$ is shown in Equation 4. Where $W_{ap}$ is the weight of the edge between the central node $a$ and the neighbor node $p$, and $S_a$ is the sum of the weights of the central node $a$ and all its neighbors.

For node $a$, the discount factor $discfactor(p)$ of its contribution to node $p$ is calculated as Equation 5, $k_p$ is the modulus length of the set of neighbors of node $p$.

The similarity $CDSim(p,q)$ between node $p$ and node $q$ can be calculated by Equation 6, where the common neighbor set of node $p$ and node $q$ is denoted by $C$.

**(5) LRE**

Zhang et al. (2018) proposed a Local Relative Entropy (LRE) method based on relative entropy and the local structure of each node to measure the similarity of node structure. It quantifies the local network structure characteristics of each node by Equation 7, where $D(k)$ is the degree of the neighbor $k$ of the central node $i$, and $m$ is the degree value of the node with the maximum degree in the network plus 1.

$$S(a,b) = |\Gamma(a) \cap \Gamma(b)| \qquad (1)$$

$$S(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{|\Gamma(a) \cup \Gamma(b)|} \qquad (2)$$

$$S(a,b) = \frac{|\Gamma(a) \cap \Gamma(b)|}{\sqrt{|\Gamma(a)||\Gamma(b)|}} \qquad (3)$$

$$ctrb(a,p) = \frac{W_{ap}}{S_a} \qquad (4)$$

Equation 8 calculates the relative entropy between different pairs of nodes to measure the difference between the structural information. Finally, the difference value is transformed into the similarity of nodes in the complex network.

**(6) RE**

Wen et al. (2019) proposed a Relative Entropy (RE) method based on Tsallis entropy. When characterizing the structural information of a node, not only the local dimension of each node is considered, but also the fractal dimension of the entire network. Equation 9 is the process of calculating the fractal dimension $d_f$ by box covering algorithm, and $N(s)$ is the number of boxes required when the box size is $s$.

The Tsallis entropy $P_{RE}(\mathrm{P}(i)||\mathrm{P}(j))$ between node $i$ and node $j$ is calculated by Equation 10, and the difference value of the information of the two nodes is then obtained. Similar to the LRE method, the difference is finally transformed into similarity to judge the similarity between nodes.

## 3.0 Data

### 3.1 Data source and preprocessing

The pre-Qin classics corpus (Li et al. 2013) we used was constructed by the Institute of Language Science and Technology of Nanjing Normal University, including 25 pre-Qin classics such as "楚辞(ChuCi)", "论语(Analects of Confucius)", "孟子(Mencius)", "孙子兵法(The Art of War by Sun Tzu)", "左传(ZuoZhuan)", etc. The corpus has been manually word-segmented, and manual part-of-speech tagging and machine-aided proofreading have been completed. The pre-Qin classics corpus is stored in database with sentences as the unit. Figure 1 is an example of some texts extracted from book chapter "楚辞·离骚(ChuCi·LiSao)".

All words in the pre-Qin classics corpus are organized as "word + part-of-speech", and a space separates the two words. For example, a sentence in "楚辞·离骚(ChuCi·LiSao)" : "*乘騏驥以馳騁兮，來吾道夫先路。*(Ride the thousand-mile horse and gallop freely, come on, I will lead the way ahead.)" This sentence, after word segmentation and part of speech tagging, becomes "*乘/v 騏驥/nx 以/p 馳/v 騁/v 兮/y ，/w 來/v 吾/r 道/v 夫/r 先/a 路/n 。/w*".

After performing basic operations such as word segmentation and statistics on the pre-Qin classics corpus, we obtained a total of 45,562 unique words (to distinguish parts of speech), and the total number of words contained in the entire corpus was 1,579,565 (including punctuation). We performed the following three steps of data cleaning operation to remove words with no actual meaning and punctuation, which are unnecessary to count similar words.

(1) Delete punctuation marks. There are no punctuation marks in the original written representation of ancient Chinese texts. Therefore, it is necessary to delete the punctuation marks that are artificially added for the convenience of reading. In order to avoid losing inter-sentence separation information during processing, we first store the classic text into a list on a sentence-by-sentence basis. Then, we match the words with the part-of-speech tag "/w" in each sentence to delete all punctuation marks.

(2) Delete words with no actual meaning. In ancient Chinese, function words (empty words) often lack specific lexical meanings, such as conjunctions like "而/c (and)", particles like "之/u (of)", and modal particles like "兮/y (exclama-

$$discfactor(p) = \frac{1}{\log(1 + k_p)} \qquad (5)$$

$$CDSim(p,q) = \sum_{a \in C} ctrb(a,p) discfactor(p) \sum_{a \in C} ctrb(a,q) discfactor(q) \qquad (6)$$

$$p(i,k) = \begin{cases} \dfrac{D(k)}{\sum_{k=1}^{m} D(k)} & k \le Degree(i) + 1 \\ 0 & k > Degree(i) + 1 \end{cases} \qquad (7)$$

$$D_{KL}(P(i)||\mathrm{P}(j)) = \sum_{k=1}^{m} p(i,k) \ln \frac{p(i,k)}{p(j,k)} \qquad (8)$$

$$d_f = -\lim_{s \to 0} \frac{\ln N(s)}{\ln s} \qquad (9)$$

$$P_{RE}(\mathrm{P}(i)||\mathrm{P}(j)) = k \sum_{k=1}^{m} \frac{\left(\frac{p_i(k)}{p_j(k)}\right)^{d_f} - \left(\frac{p_i(k)}{p_j(k)}\right)}{1 - d_f} \qquad (10)$$

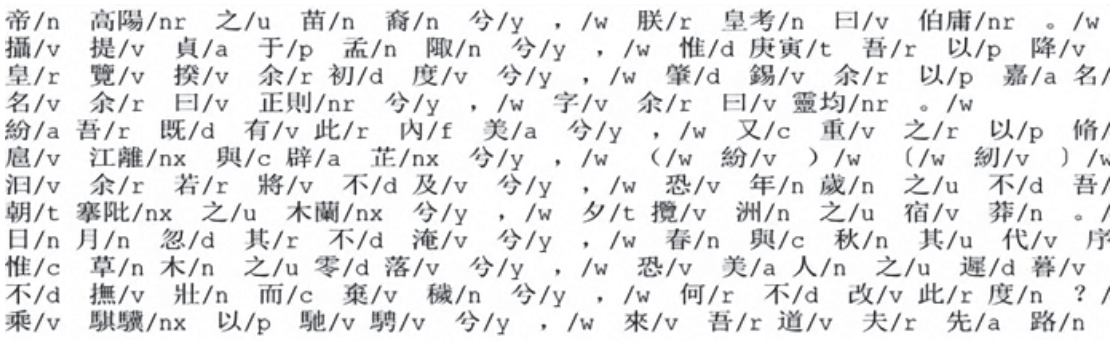
帝/n 高陽/nr 之/u 苗/n 裔/n 兮/y ，/w 朕/r 皇考/n 曰/v 伯庸/nr 。/w
攝/v 提/v 貞/a 于/p 孟/n 陬/n 兮/y ，/w 惟/d 庚寅/t 吾/r 以/p 降/v
皇/r 覽/v 揆/v 余/r 初/d 度/v 兮/y ，/w 肇/d 錫/v 余/r 以/p 嘉/a 名/
名/v 余/r 曰/v 正則/nr 兮/y ，/w 字/v 余/r 曰/v 靈均/nr 。/w
紛/a 吾/r 既/d 有/v 此/r 內/f 美/a 兮/y ，/w 又/c 重/v 之/r 以/p 脩/
扈/v 江離/nx 與/c 辟/a 芷/nx 兮/y ，/w （/w 紛/v ）/w （/w 紉/v ）/w
汩/v 余/r 若/r 將/v 不/d 及/v 兮/y ，/w 恐/v 年/n 歲/n 之/u 不/d 吾/
朝/t 搴阰/nx 之/u 木蘭/nx 兮/y ，/w 夕/t 攬/v 洲/n 之/u 宿/v 莽/n 。/
日/n 月/n 忽/d 其/r 不/d 淹/v 兮/y ，/w 春/n 與/c 秋/n 其/u 代/v 序
惟/c 草/n 木/n 之/u 零/d 落/v 兮/y ，/w 恐/v 美/a 人/n 之/u 遲/d 暮/v
不/d 撫/v 壯/n 而/c 棄/v 穢/n 兮/y ，/w 何/r 不/d 改/v 此/r 度/n ？/
乘/v 騏驥/nx 以/p 馳/v 騁/v 兮/y ，/w 來/v 吾/r 道/v 夫/r 先/a 路/n

*Figure 1*. Example of pre-Qin classics corpus.

| No. | Word | Freq. | No. | Word | Freq. |
|---|---|---|---|---|---|
| 1 | 之/r | 22677 | 11 | 君/n | 5954 |
| 2 | 其/r | 20930 | 12 | 民/n | 5536 |
| 3 | 曰/v | 19136 | 13 | 是/r | 5031 |
| 4 | 者/r | 16204 | 14 | 國/n | 4798 |
| 5 | 有/v | 12283 | 15 | 何/r | 4781 |
| 6 | 為/v | 11888 | 16 | 此/r | 4520 |
| 7 | 人/n | 10659 | 17 | 使/v | 4395 |
| 8 | 無/v | 6215 | 18 | 能/v | 4015 |
| 9 | 可/v | 6101 | 19 | 得/v | 3954 |
| 10 | 所/r | 6053 | 20 | 謂/v | 3919 |

*Table 1.* Basic statistics of the top 20 high-frequency words in the corpus after cleaning

tory)". However, these words usually appear frequently, which will interfere with the similarity calculation results. Therefore, we treat them as a stop words and delete them. In addition, through manual screening, nouns and verbs with no apparent meaning were filtered.

(3) Delete the isolated words. A word not associated with any word will eventually become an isolated node in the language network, and its similarity cannot be calculated. For example, the sentence "*倦/v 也/y 。/w* (This is tired.)", after deleting punctuation marks and nonsense words, becomes an independent vocabulary "倦/v (tired)".

After the above processing, the final corpus contains 42,916 non-repetitive words (to distinguish parts of speech), and the total number of words is 921,344. Table 1 shows the top 20 high-frequency words after processing, where "Freq." refers to the frequency of word occurrence.

### 3.2 Construction of the Pre-Qin Classics Language Network

In order to avoid the interference of "multi-part-of-speech of a word" on the similarity calculation, we use the combination of "word + part-of-speech" as the node in the Pre-Qin classics language network (hereinafter referred to as PQLN). An edge is established between the nodes represented by two adjacent words in the sentence. The weight of the edge is numerically equal to the number of times the two nodes appear together in the corpus.

We use Pajek5.08 software to construct PQLN. The basic statistical data of the weighted undirected network is as follows: the number of nodes is 42,916, and there are 294,236 edges in total. Among them, there are 63,314 edges with a weight of 1, accounting for 21.52% of the total number of edges. PQLN's overall betweenness centrality (BC) is 0.1025, the clustering coefficient is 0.0003, the average degree of nodes is 13.71, and the network diameter is 18. The node "曰

/v (say)" has the most significant degree with a value of 5,644. The degrees of "其/r (his/her)" and "有/v (have)" are ranked second and third. These three words are also the top three in BC, ranked in the top 5 of the high-frequency words in Table 1. Since the PQLN network is too large to be directly presented, we used part of the texts from "孙子兵法(The Art of War by Sun Tzu)" to construct a small language network for visual display; see Figure 2 below.

### 4.0 Proposed methods

Among the above-mentioned existing node similarity calculation methods for complex network, the methods based on common neighbor nodes (CN, Jaccard, Salton and CDSim) believe that the more common neighbors of two nodes, the more likely they are to be similar. However, when two nodes do not have a direct common neighbor, the similarity between them will not be calculated, which quickly leads to the omission of potentially similar words. The algorithms based on relative entropy (LRE, RE), although calculating the similarity between all nodes in the network, ignore the weight information of the edges between nodes, and treat the importance of neighbors to the central node equally. In fact, due to the different frequency of co-occurrence of words, the amount of information between different neighbors and the central node is often different. For example, the node "曰/v" in Figure 2 is directly connected to both "孫子/nr (Sun Tzu)" and "道/n (Tao/way)". However, the amount of information between "道/n" and "曰/v" is significantly larger and should be given greater weight in the calculation process. Therefore, we improved the algorithm of Zhang et al. (2018) and proposed a weighted language network node similarity calculation method (LREW). We take the network shown in Figure 3 as an example to help illustrate the proposed method.

### 4.1 Build probability sets

First, we establish a probability set to represent the local network structure information of each node. Define the local network of a node consists of a central node and its first-order neighbor nodes. For node $i$, its probability set is denoted as $P(i)$. In order to be able to perform relative entropy calculations, it is necessary to ensure that the set lengths are consistent. Suppose the length of the set is $m$, and its value is equal to the maximum degree $D_{max}$ of the nodes in the network plus 1 (Equation 11). In the network

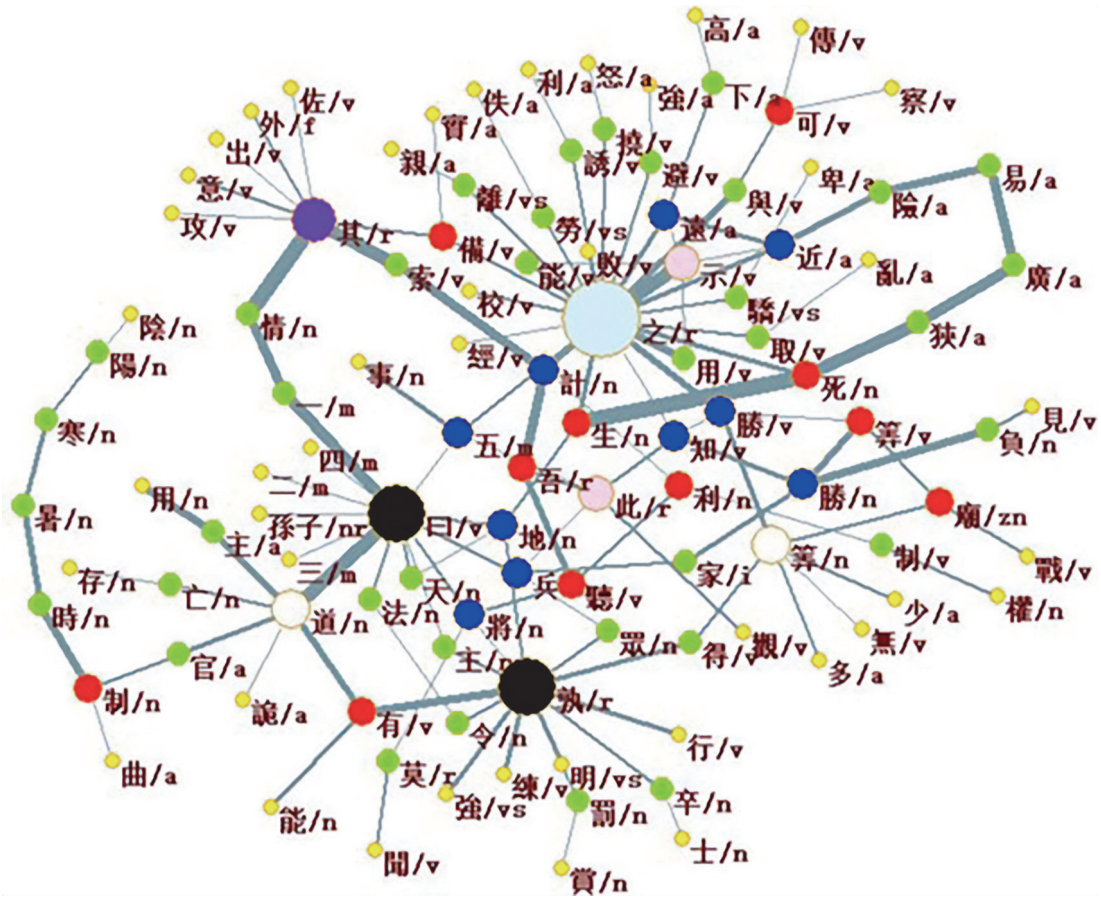


*Figure 2.* The language network constructed by some texts of "The Art of War by Sun Tzu".

shown in Figure 3, the value of $m$ is 8. $P(i)$ is expressed by Equation 12.

For each element $p(i,k)$ in the probability set, $i$ represents the central node and $k$ represents the neighbor. $p(i,k)$ is the amount of structural information contributed by node $k$ to node $i$. The structural information is composed of the degree of the node and the strength of the connection between the node pairs. Define the degree of node $i$ as $D(i)$, then the sum of degrees $D_{sum}(i)$ of all nodes in the local network is calculated as follows.

In Equation 13, $k$ is the first-order neighbor of the node $i$, and $D(k)$ is the degree of $k$. Take the node "百/m (hundred)" and node "五十/m (fifty)" in Figure 3 as an example, $D_{sum}(百/\mathrm{m}) = 3+2+2+5 = 12$, $D_{sum}(五十/\mathrm{m}) = 3+2+2+5 = 12$. The weight of the edge is used to express the strength of the connection between the node pairs in the local network. Define the sum of local network weights as $W_{sum}(i)$, and its calculation is as Equation 14.

$W(i,k)$ is numerically equal to the weight of the edge between the central node $i$ and the neighbor $k$. In Figure 3, the value of the edge weight between nodes has been marked with numbers. $W_{sum}(百/\mathrm{m}) = 1+3+7 = 11$, $W_{sum}(五十/\mathrm{m}) = 1+1+3 = 5$. Since the degree of most nodes in the network is less than $D_{max}$, Equation 15 is defined when calculating the amount of structural information $p(i,k)$ contributed by node $k$ to the central node i.

When node $k$ is not the central node $i$, ① if $k$ is within the range of first-order neighbors, define $p(i,k)$ as the product of the amount of information contributed by node $k$ in terms of node degree and weight. Among them, the degree information of a node is the ratio of $D(k)$ to the sum of the degrees of all nodes in the local network $D_{sum}(i)$. The weight information is the ratio of the edge weight $W(i,k)$ to the sum of the weight $W_{sum}(i)$ of the local network. ② If $k$ exceeds the range of the first-order neighbors, all the remaining $m - D(i) - 1$ elements in the probability set $P(i)$ are as-

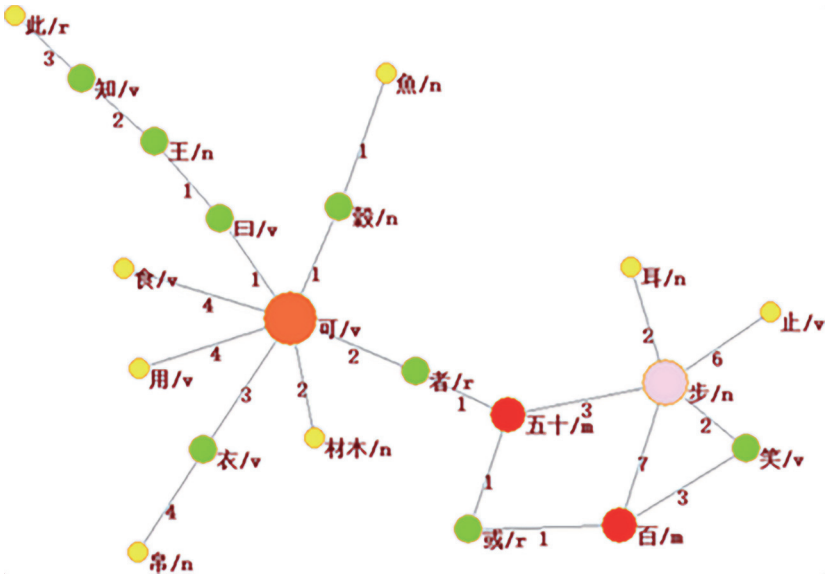


*Figure 3.* The language network constructed by some texts of "Mencius".

$$m = D_{max} + 1 \qquad (11)$$

$$P(i) = [p(i,1), p(i,2), \ldots, p(i,k), \ldots, p(i,m)] \qquad (12)$$

$$D_{sum}(i) = D(i) + \sum_{k=1}^{D(i)} D(k) \qquad (13)$$

$$W_{sum}(i) = \sum_{k=1}^{D(i)} W(i,k) \qquad (14)$$

signed a value of 0. When $k$ is the central node $i$, since the central node can be regarded as the common neighbor node of the rest of the nodes in the local network, the node $i$ is defined to share the weight of each neighbor, and its weight $W(i,i)$ is numerically equivalent to $W_{sum}(i)$. Therefore,

$P(\text{百/m}) = \left[\frac{2\times1}{12\times11}, \frac{2\times3}{12\times11}, \frac{5\times7}{12\times11}, \frac{3}{12}, 0,0,0,0\right] = \left[\frac{1}{66}, \frac{1}{22}, \frac{35}{132}, \frac{1}{4}, 0,0,0,0\right]$, $P(\text{五十/m}) = \left[\frac{2\times1}{12\times5}, \frac{2\times1}{12\times5}, \frac{5\times3}{12\times5}, \frac{3\times5}{12\times5}, 0,0,0,0\right] = \left[\frac{1}{30}, \frac{1}{30}, \frac{1}{4}, \frac{1}{4}, 0,0,0,0\right]$.

### 4.2 Calculate relative entropy

Relative entropy (KL divergence) is a measure commonly used to calculate the difference between two probability distributions. We learn from the ideas of Zhang et al. (2018) and convert the similarity between nodes into a difference calculation. Since the order of the elements in the probability set $P(i)$ will affect the final relative entropy calculation result, in order to make each node have only one fixed probability expression, we process the elements in the probability set in descending numerical order. The probability set after descending order processing is represented by $P'(i)$, and each element is represented by $p'(i,k)$. $P'(\text{百/m}) = \left[\frac{35}{132}, \frac{1}{4}, \frac{1}{22}, \frac{1}{66}, 0,0,0,0\right]$, $P'(\text{五十/m}) = \left[\frac{1}{4}, \frac{1}{4}, \frac{1}{30}, \frac{1}{30}, 0,0,0,0\right]$.

For node $i$ and node $j$, the relative entropy $P_{LREW}(P'(i)||P'(j))$ is calculated as follows (Equation 16).

Since $p'(i,k)$ and $p'(j,k)$ may have a value of 0, define the threshold $d$, which is calculated by selecting the node $i$ and node $j$ with the smaller degree, and add 1 to the degree value to ensure that all elements in the probability set participating in the calculation are not 0 (Equation 17).

Here, the value of $d$ is 4 and $P_{LREW}(P'(\text{百/m})||P'(\text{五十/m})) = \frac{1}{4}\times\ln\frac{\frac{1}{4}}{\frac{1}{4}} + \frac{35}{132}\times\ln\frac{\frac{35}{132}}{\frac{1}{4}} + \frac{1}{22}\times\ln\frac{\frac{1}{22}}{\frac{1}{30}} + \frac{1}{66}\times\ln\frac{\frac{1}{66}}{\frac{1}{30}} = 0.0177$,

$P_{LREW}(P'(\text{五十/m})||P'(\text{百/m})) = \frac{1}{4}\times\ln\frac{\frac{1}{4}}{\frac{1}{4}} + \frac{1}{4}\times\ln\frac{\frac{1}{4}}{\frac{35}{132}} + \frac{1}{30}\times\ln\frac{\frac{1}{30}}{\frac{1}{22}} + \frac{1}{30}\times\ln\frac{\frac{1}{30}}{\frac{1}{66}} = 0.0012$.

### 4.3 Calculate similarity

We use the method of Wen et al. (2019) to convert the relative entropy difference into similarity. Based on Equation 16, the values of $P_{LREW}(P'(i)||P'(j))$和$P_{LREW}(P'(j)||P'(i))$ are not the same. In order to make the difference value between two nodes unique, we adopt the following Equation 18, using $r(i,j)$ to represent the difference in the amount of information between node $i$ and node $j$.

For the node "百/m (hundred)" and node "五十/m (fifty)", $r(\text{百/m}, \text{五十/m}) = P_{LREW}(P'(\text{百/m})||P'(\text{五十/m})) + P_{LREW}(P'(\text{五十/m})||P'(\text{百/m})) = 0.0177 + 0.0012 = 0.0189$. The information amount difference between every two node pairs is normalized, and the ratio is calculated with the maximum information amount difference $\max(r(i,j))$ in the network. Through Equation 19, the difference in information amount is converted into the similarity between nodes, so as to obtain the similarity $S(i,j)$ between node $i$ and node $j$.

Finally, construct the similarity matrix $S$. For each element $S(i,j)$ in the similarity matrix, the value range is [0,1]. The closer the value is to 1, the smaller the difference in the amount of information between the words represented by node $i$ and node $j$, and the higher the similarity. On the contrary, the greater the difference, the lower the similarity (Equation 20).

We used the proposed LREW algorithm to calculate the similarity of the nodes in the network shown in Figure 3. Table 2 lists the top 5 nodes with the highest similarity scores for each node, where "S1-S5" represents the top 5 most similar words in descending order of similarity.

$$p(i,k) = \begin{cases} \dfrac{D(k)W(i,k)}{D_{sum}(i)W_{sum}(i)} & k \neq i, k \leq D(i)+1 \\ 0 & k \neq i, k > D(i)+1 \\ \dfrac{D(k)}{D_{sum}(i)} & k = i \end{cases} \quad (15)$$

$$P_{LREW}(P'(i)||P'(j)) = \sum_{k=1}^{d} p'(i,k)\ln\frac{p'(i,k)}{p'(j,k)} \quad (16)$$

$$d = \min\big(D(i), D(j)\big) + 1 \quad (17)$$

| No. | Word | S1 | S2 | S3 | S4 | S5 |
|---|---|---|---|---|---|---|
| 1 | **百/m** | 五十/m | 衣/v | 穀/n | 曰/v | 王/n |
| 2 | **五十/m** | 百/m | 衣/v | 穀/n | 曰/v | 王/n |
| 3 | **耳/n** | 止/v | 材木/n | 用/v | 食/v | 魚/n |
| 4 | **王/n** | 曰/v | 者/r | 知/v | 衣/v | 穀/n |
| 5 | **曰/v** | 王/n | 者/r | 衣/v | 穀/n | 知/v |
| 6 | **帛/n** | 此/r | 魚/n | 止/v | 耳/n | 食/v |
| 7 | **穀/n** | 衣/v | 者/r | 曰/v | 百/m | 五十/m |
| 8 | **用/v** | 食/v | 材木/n | 止/v | 耳/n | 魚/n |
| 9 | **或/r** | 笑/v | 王/n | 曰/v | 知/v | 者/r |
| 10 | **此/r** | 魚/n | 帛/n | 止/v | 耳/n | 食/v |
| 11 | **可/v** | 步/n | 者/r | 穀/n | 衣/v | 曰/v |
| 12 | **者/r** | 知/v | 曰/v | 穀/n | 王/n | 衣/v |
| 13 | **知/v** | 者/r | 曰/v | 王/n | 穀/n | 衣/v |
| 14 | **笑/v** | 或/r | 王/n | 曰/v | 知/v | 衣/v |
| 15 | **衣/v** | 穀/n | 百/m | 曰/v | 五十/m | 者/r |
| 16 | **魚/n** | 此/r | 帛/n | 止/v | 耳/n | 食/v |
| 17 | **材木/n** | 食/v | 用/v | 止/v | 耳/n | 魚/n |
| 18 | **止/v** | 耳/n | 材木/n | 用/v | 食/v | 魚/n |
| 19 | **步/n** | 可/v | 者/r | 穀/n | 衣/v | 曰/v |
| 20 | **食/v** | 材木/n | 用/v | 止/v | 耳/n | 魚/n |

*Table 2:* Example network node similarity calculation results.

$$r(i,j) = P_{LREW}(P'(i)||P'(j)) + P_{LREW}(P'(j)||P'(i)) \qquad (18)$$

$$S(i,j) = 1 - \frac{r(i,j)}{\max(r(i,j))} \qquad (19)$$

$$S = \begin{bmatrix} S(1,1) & \cdots & S(1,j) \\ \vdots & \ddots & \vdots \\ S(i,1) & \cdots & S(i,j) \end{bmatrix} \qquad (20)$$

## 5.0 Comparative experiment

### 5.1 Overall processing flow

The overall process of the node similarity calculation experiment is shown in Figure 4. First of all, 25 pre-Qin classics were preprocessed to construct the PQLN. Secondly, compare the similar word mining performance on PQLN between our proposed LREW method and other six node similarity calculation methods. Finally, three indicators of DIRECT, P(n) and INVR are used to evaluate the overall performance of the seven algorithms, and the experimental results are comprehensively analyzed.

### 5.2 Task word selection

In the pre-Qin classics corpus, nouns account for 66.50% (28,539 words), verbs account for 18.84% (8,084 words), and the two parts of speech together account for 85.34% of all words. Therefore, we selected nouns and verbs as task words for analysis. In order to more comprehensively reflect the performance of each algorithm, we invited experts from the School of Liberal Arts to screen and determine the task words. During the selection process, apart from considering the representativeness of the word meanings, it is also necessary to cover different frequency distribution intervals as much as possible. Considering the characteristics of word formation in ancient Chinese, we selected not only mono-

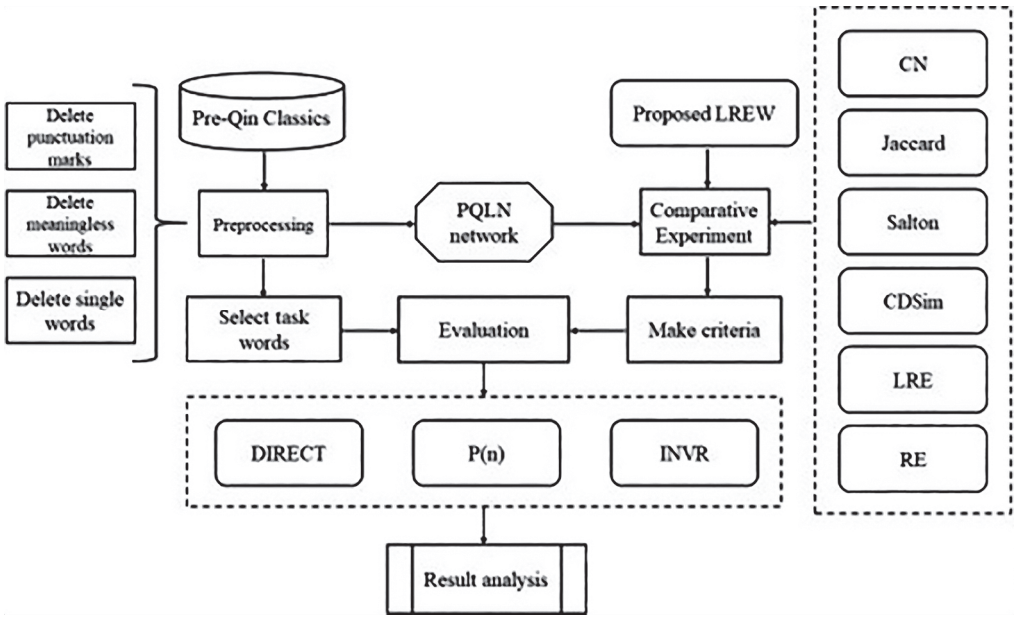


*Figure 4.* Experiment process of node similarity calculation of PQLN.

syllabic words but also some disyllabic words. Since we will manually score similar words for each selected target word in Section 5.3, taking into account both the vocabulary coverage and human labor cost, following the approach of previous research (Han et al. 2015), we finally selected 40 task words. The results are shown in Table 3, where "Freq." refers to the frequency of word occurrence.

### 5.3 Similarity Evaluation Method

We use the evaluation indicators designed by Curran and Moens (2002), combined with two aspects of manual evaluation criteria formulation and algorithm evaluation, to evaluate the performance of node similarity calculation methods. For the evaluation criteria, firstly, we comprehensively use authoritative ancient Chinese dictionaries such as "Dictionary of Commonly Used Ancient Chinese Synonyms", "Dictionary of Ancient Chinese", and "Chinese Classics", and select up to 30 similar words as candidate words for each task word. Next, three experts with a background in ancient Chinese literature are invited to sort, add or delete the provided candidate words, and give the first 30 candidate words most similar to each task word in sequence. Then take the inverse of the rank order of each word according to three experts and add them together, and arrange the final scores in descending order, and select the top 20 words as the evaluation criteria for similarity results.

We select the Direct Match (DIRECT), the P(n) and the INVR method to evaluate the performance of the seven algorithms. The following is a brief introduction to the implementation process of these three evaluation methods.

(1) Direct Match (DIRECT). This method directly counts the number of hits in the evaluation criteria of the Top-N words given by the similarity algorithm, and the value of N is the number of similar words in the evaluation criteria. Sum the number of hits to get the DIRECT score.

(2) P(n). Compared with the DIRECT method, this method can flexibly select the top-n words recommended by the similarity algorithm for performance evaluation. If the number of words hit is m, then $\mathrm{P(n)} = m/n$.

(3) Inverse Rank (INVR). This method also examines the hits in the evaluation criteria for the first N results given by the similarity algorithm. Different from the simple counting of the DIRECT method, the score of the INVR method is the sum of the reciprocal of the rank where the hit words are found.

### 5.4 Comparative analysis of noun similarity

Table 4 shows the top 20 words most similar to "君/n (monarch)" based on the LREW algorithm and other six node similarity calculation methods. By querying authoritative resources such as "The ancient Chinese Dictionary of Common Synonyms" and "Ancient Chinese Dictionary", words with similar relations have been marked in bold.

Among the four algorithms based on common neighbor, the Salton, Jaccard, CDSim and CN methods identified 6, 5, 4, and 3 words similar to "君/n" respectively. The best performing Salton method identified the synonyms "王/n (king)" and "主/n (lord)" with the meaning of "君/n" as the ruler of a country, and the corresponding words "臣/n (min-

| Noun | | | | Verb | | |
|---|---|---|---|---|---|---|
| **No.** | **Word** | **Freq.** | **Rank** | **Word** | **Freq.** | **Rank** |
| 1 | 君/n | 5954 | 11 | 曰/v | 19136 | 3 |
| 2 | 國/n | 4798 | 14 | 為/v | 11888 | 6 |
| 3 | 天下/n | 2879 | 34 | 使/v | 4395 | 17 |
| 4 | 諸侯/n | 1981 | 51 | 行/v | 3116 | 28 |
| 5 | 罪/n | 1148 | 115 | 聞/v | 2060 | 48 |
| 6 | 車/n | 840 | 163 | 及/v | 1567 | 83 |
| 7 | 惡/n | 494 | 290 | 亡/v | 1299 | 100 |
| 8 | 邑/n | 469 | 314 | 食/v | 1230 | 107 |
| 9 | 百姓/n | 451 | 328 | 好/v | 840 | 164 |
| 10 | 社稷/n | 307 | 498 | 惡/v | 817 | 170 |
| 11 | 玉/n | 302 | 503 | 克/v | 496 | 287 |
| 12 | 祭/n | 287 | 522 | 云/v | 352 | 448 |
| 13 | 有司/n | 209 | 723 | 弒/v | 322 | 480 |
| 14 | 勝/n | 188 | 771 | 之/v | 251 | 604 |
| 15 | 病/n | 168 | 847 | 赦/v | 161 | 876 |
| 16 | 賓客/n | 89 | 1363 | 蔽/v | 156 | 904 |
| 17 | 中國/n | 82 | 1440 | 配/v | 75 | 1528 |
| 18 | 邦國/n | 72 | 1565 | 貶/v | 63 | 1705 |
| 19 | 宰夫/n | 56 | 1858 | 偷/v | 60 | 1762 |
| 20 | 庶民/n | 49 | 2041 | 乏/v | 41 | 2333 |

*Table 3.* Basic information of task words.

ister)" and "民/ n (civilian)". It also found the synonym "公/n (an honorific name referring to 'you')" which takes the meaning of appellation, and the hypernym "人/n (people)". For relative entropy-based algorithms, the number of similar words identified by the LRE, RE and our proposed LREW is 2, 5 and 3 respectively. The three algorithms all identified the similar word "民/n". The remaining 4 similar words identified by the RE method were all names, and the similar words detected by LRE and LREW were all common nouns. For the recognition of similar words of "君/n", the method based on relative entropy is slightly inferior to the method based on common neighbors. However, our proposed LREW method incorporating the feature of edge weights between local network node pairs, identified more similar words compared to that of the LRE method. It also outperformed the RE method in recognizing similar words for common nouns.

Table 5 lists the similarity calculation results of the target word "齊/ns (Qi, a place name)". For each method, the top 20 recommended words are listed in descending order of similarity. The words that genuinely exhibit similarity with the target word are highlighted in bold.

The Jaccard, Salton and CDSim methods all identified 7 similar words, while the CN method did not hit any similar words. Our proposed LREW method identified 3 similar words, and the LRE and RE methods identified 2 similar words respectively. Combined with the results in Table 4, it shows that there are still deficiencies in the extraction of node information among the three relative entropy-based algorithms. The LREW method outperforms the LRE in both the number of recognized similar words and the ranking of the first similar words, which shows that the weight information does contain important features. Additionally, we can observe that the LREW method exhibits higher calculation accuracy for similar words of place names.

### 5.5 Comparative analysis of verb similarity

Table 6 lists the first 20 similar words of the target word "如/v (towards)" identified by the similarity calculation methods. Similarly, after comparison with authoritative sources, the actual similar words are marked in bold.

In ancient Chinese, "如/v" has the following meanings: ① obey. ②Towards, go to... ③ Like, similar to... ④ Can reach, comparable. ⑤ Yes, should. ⑥ How. Within the al-

| No. | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| 1 | 君/n | 君/n | 君/n | 君/n | 君/n | 君/n | 君/n |
| 2 | 其/r | **人/n** | **人/n** | 之/r | 所/r | 是/r | 得/v |
| 3 | 之/r | 可/v | **臣/n** | 其/r | 是/r | **民/n** | 立/v |
| 4 | 者/r | 所/r | 可/v | 者/r | **民/n** | 所/r | 國/n |
| 5 | 曰/v | **臣/n** | 所/r | **人/n** | 能/v | 謂/v | 所/r |
| 6 | 有/v | 是/r | 國/n | 可/v | 使/v | 殉/v | 見/v |
| 7 | 為/v | 能/v | 是/r | 所/r | 可/v | 能/v | 一/m |
| 8 | **人/n** | **民/n** | 能/v | 國/n | 得/v | **丘/nr** | 三/m |
| 9 | 可/v | 國/n | **民/n** | **臣/n** | 無/v | 殯/n | 事/n |
| 10 | 無/v | 吾/r | 吾/r | **民/n** | 我/r | **季武子/nr** | 行/v |
| 11 | 所/r | **王/n** | 子/r | 能/v | **人/n** | 喻/v | 何/r |
| 12 | **民/n** | 我/r | **王/n** | 是/r | 謂/v | 邇/n | 生/v |
| 13 | 是/r | 何/r | **主/n** | 曰/v | 國/n | 成/a | **子/n** |
| 14 | 能/v | 此/r | 之/r | 吾/r | 一/m | **鮑叔/nr** | 卒/v |
| 15 | 使/v | 得/v | 此/r | 有/v | 何/r | **惠子/nr** | **民/n** |
| 16 | 國/n | **公/n** | 其/r | 子/r | 大/a | 由/n | 作/v |
| 17 | 我/r | 德/n | **公/n** | 無/v | 如/v | 鉶/n | 欲/v |
| 18 | 得/v | 事/n | 者/r | **公/n** | 三/m | 因/n | 五/m |
| 19 | 何/r | 天下/n | 何/r | 何/r | 知/v | 內/a | 死/v |
| 20 | **臣/n** | 子/r | 我/r | 此/r | 此/r | 達/n | **臣/n** |

*Table 4.* Word similarity calculation results of the target word "君/n (monarch)".

gorithms based on common neighbors, only the Jaccard method recognized a similar word "行/v (go)", and its ranking is 20th. The other three algorithms failed to provide correct similar words in the top 20 results. The relative entropy-based methods performed relatively well. The LRE method also recognizes the similar word "行/v", and the similarity is ranked 10th, which is higher than that of the Jaccard method. Our proposed LREW method recognizes three similar words "至/v (go)", "來/v (come)", and "行/v", and its performance is significantly better than all other 6 algorithms. This also shows that adding the weight information between nodes can indeed improve the algorithm.

Table 7 shows the recognition results of similar words for the target word "克/v (overcome)".

Similarly, the top 20 similar words are selected and arranged in descending order of similarity.

No similar words of "克/v" were hit in the top 20 results of the CN method. Among the other three common neighbor-based algorithms, the Jaccard method performed best, identifying 6 similar words. However, the CDSim method with weighted features only recognized one similar word, and its performance is not as good as the algorithm that only considers node degree information. Among the relative entropy-based methods, our proposed LREW method performed best, identifying 4 similar words, and both LRE and RE methods identified 2 similar words. The rank of the first similar word identified by the LREW method is also higher than the other two relative entropy-based methods.

### 5.6 Overall similarity evaluation

Through the comparative analysis of two nouns and verbs, we can roughly understand the performance of each method. We further use the DIRECT, P(n) and INVR method to evaluate the overall performance of our proposed LREW and other six methods. The DIRECT method consists of two indicators, the sum and average value, and the P(n) method compares the results under the three levels of n as 1, 5, and 10. The overall evaluation results of nouns and verbs are shown in Table 8 and Table 9 respectively. Among the algorithms in the common neighbor-based group and the relative entropy-based group, the highest-scoring indicators are marked in bold.

For nouns in the PQLN network, among the four methods based on common neighbors, the CDSim method achieved the best similar word recognition performance. Among the three methods based on relative entropy, our proposed LREW method achieved (or tied) the highest scores in all indicators. The CDSim method and the LREW method

| No. | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| 1 | 齊/ns | 齊/ns | 齊/ns | 齊/ns | 齊/ns | 齊/ns | 齊/ns |
| 2 | 曰/v | **晉/ns** | **晉/ns** | **宋/ns** | 聽/v | **楚/ns** | 百/m |
| 3 | 之/r | **楚/ns** | **楚/ns** | **晉/ns** | 命/n | 居/v | **楚/ns** |
| 4 | 者/r | **宋/ns** | **宋/ns** | **楚/ns** | 行/n | 食/v | 師/n |
| 5 | 其/r | 公/n | 公/n | **鄭/ns** | **楚/ns** | 諸/j | **晉/ns** |
| 6 | 有/v | 諸侯/n | **魯/ns** | 之/r | **晉/ns** | 下/f | 東/f |
| 7 | 為/v | **魯/ns** | 諸侯/n | **魯/ns** | 下/f | 君子/n | 天/n |
| 8 | 人/n | **鄭/ns** | **秦/ns** | 何/r | 舉/v | 聽/v | **宋/ns** |
| 9 | 使/v | 子/r | **鄭/ns** | 公/n | 多/a | 時/n | 車/n |
| 10 | 可/v | **秦/ns** | 國/n | 君/n | 事/v | 治/v | 自/r |
| 11 | 無/v | 師/n | 子/r | 曰/v | 兵/n | **晉/ns** | 日/n |
| 12 | 君/n | 故/n | **吳/ns** | 諸侯/n | 食/v | 身/n | 諸侯/n |
| 13 | 所/r | 國/n | 君/n | **衛/ns** | 政/n | 命/v | 馬/n |
| 14 | 是/r | 王/n | 故/n | 國/n | 世/n | 事/v | 士/n |
| 15 | 國/n | 吾/r | 師/n | 可/v | 外/f | 上/n | 外/f |
| 16 | 能/v | 此/r | 此/r | 王/n | 小/a | 邇/n | 世/n |
| 17 | 民/n | **吳/ns** | 王/n | 所/r | 爾/r | 爲/v | 成/v |
| 18 | 何/r | 地/n | 吾/r | **曹/ns** | 功/n | 法/n | 者/n |
| 19 | 我/r | 眾/n | 何/r | 者/r | 居/v | 喻/v | 來/v |
| 20 | 此/r | 今/t | 衛/ns | 使/v | 諸/j | 多/a | 內/f |

*Table 5.* Word similarity calculation results of the target word "齊/ns (Qi, a place name)".

both include the weight information between the node pairs. This shows that for the pre-Qin classics, the co-occurrence frequency of words can indeed better supplement the structural information of the local network, thus highlighting similarity. Compared with the original LRE method, the improved LREW method performs better in terms of the absolute number of similar words recognized and the ranking of the first 20 words. This also shows that the weight information further refines the structural characteristics of the local network, making the recognition of similar words more accurate.

Similarly, in the overall recognition performance of 20 verbs, the CDSim and our proposed LREW method both achieved the highest scores in their groups respectively. Compared with the evaluation results of nouns, all seven algorithms exhibit better recognition performance for verbs. This shows that in PQLN, the local network of verbs is more distinguishable.

Although our proposed LREW is superior to the other relative entropy-based similarity algorithms, it still falls short when compared to the common neighbor-based algorithms. The reason is that the common neighbor-based methods rely on the premise of having shared neighbors between two nodes for similarity calculation. As a result, not all nodes in the network are compared, and the number of nodes participating in the calculation is reduced, making it more likely to match similar words and rank them higher in the results. Our proposed LREW method is based on relative entropy. Although this method only considers local network information, it does not limit the common neighbors. Therefore, it matches and calculates all nodes in the network.

In order to verify this conclusion, we took the network constructed in Figure 3 as an example and employed the CDSim method, which is the most effective common neighbor-based approach, to calculate the similarity between all words. Table 10 lists the top 5 words in terms of similarity, and the "Count" field in the table header indicates the number of words involved in the calculation.

Among these 20 words, half of them have fewer than 5 words available for calculation, and even 4 words have only one eligible node for calculation. This dramatically reduces the number of words involved in the similarity calculation, which is equivalent to sacrificing recall to ensure precision, so it is easy to rank similar words higher. The LREW method, on the other hand, is not limited by the constraints of common neighbors and can include all the words in the network for calculation. So it can obtain more comprehensive calculation results.

| No. | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| 1 | 如/v | 如/v | 如/v | 如/v | 如/v | 如/v | 如/v |
| 2 | 曰/v | 大/a | 大/a | 者/r | 謂/v | 謂/v | 所/r |
| 3 | 其/r | 得/v | 吾/r | 其/r | 一/m | 我/r | 爲/v |
| 4 | 有/v | 一/m | 無/v | 曰/v | 我/r | 得/v | 民/n |
| 5 | 為/v | 是/r | 得/v | 為/v | 大/a | 能/v | 在/v |
| 6 | 者/r | 無/v | 一/m | 有/v | 何/r | 殉/v | **至/v** |
| 7 | 之/r | 吾/r | 人/n | 之/r | 三/m | 喻/v | 我/r |
| 8 | 無/v | 可/v | 是/r | 無/v | 知/v | 季武子/nr | 伐/v |
| 9 | 人/n | 人/n | 從/v | 君/n | 國/n | 何/r | 王/n |
| 10 | 使/v | 知/v | 可/v | 知/v | **行/v** | 殯/n | 使/v |
| 11 | 可/v | 何/r | 知/v | 人/n | 見/v | 邇/n | **來/v** |
| 12 | 是/r | 從/v | 者/r | 在/v | 用/v | 一/m | 可/v |
| 13 | 所/r | 此/r | 此/r | 從/v | 此/r | 達/n | 卒/v |
| 14 | 君/n | 三/m | 何/r | 得/v | 得/v | 因/n | 三/m |
| 15 | 得/v | 出/v | 三/m | 受/v | 出/v | 鍼/n | 用/v |
| 16 | 一/m | 所/r | 有/v | 使/v | 吾/r | 鄭國/ns | 君/n |
| 17 | 民/n | 君/n | 出/v | 可/v | 欲/v | 襄子/nr | 立/v |
| 18 | 能/v | 能/v | 禮/n | 大/a | 事/n | 荊王/nr | **行/v** |
| 19 | 大/a | 在/v | 在/v | 能/v | 王/n | 甲午/t | 見/v |
| 20 | 何/r | **行/v** | 為/v | 出/v | 從/v | 見/v | 人/n |

*Table 6.* Word similarity calculation results of the target word "如/v (towards)".

Comparing Table 3 and Table 10, we can observe that for similar words like "五十/m (fifty)" and "百/m (hundred)", both the common neighbor-based methods and our improved LREW method can calculate their similarity. However, when we look for similar words for "笑/v (laugh)" in the network, the common neighbor-based algorithm only computes the similarity between "笑/v" and six other words: "耳/n (ear)", "止/v (stop)", "步/n (step)", "或/r (or)", "五十/m", and "百/m". Among these words, only "止/v" is a verb, but its meaning is not similar to "笑/v" , so no potential similar word is found in the network.

In contrast, the LREW method involves all the remaining 19 words in the similarity calculation, ensuring a comprehensive recognition of potential similar words. From the calculation results of the top 5 similar words for "笑/v" in Table 3, "曰/v (say)", "知/v (know)", and "衣/v (clothing)" are all verbs, and intuitively, "曰/v" appears to have a higher degree of similarity with "笑/v" compared to "止/v". For large complex networks like PQLN, where the network diameter reaches 18, many nodes have no direct neighbors. Therefore, our proposed LREW method ensures that even if two words are far away from each other in the network, their similarity can still be computed, leading to more comprehensive results. But it may cause some similar words to be ranked relatively late, and thus perform poorly in the recommended top 20 similar words and overall similar word evaluation.

## 6.0 Conclusion

Oriented to the digital humanities, we proposed an improved weighted network node similarity calculation method based on local relative entropy (LREW). Unlike algorithm that only considers the degree information of nodes, this method incorporates weight information between nodes into the local feature representation of nodes. By quantifying the contribution of high-frequency words and low-frequency words to the central node, it enhances the distinction of local network structures among different nodes and improves the accuracy of similarity calculation.

The performance our proposed LREW method is better than that of LRE and RE method, and it performs best in the relative entropy-based algorithms. The LREW method is weaker than the common neighbor-based algorithms in identifying similar words in the PQLN network. This is because in the latter's algorithms implementation process, it is stipulated that two nodes must have shared neighbors to be included in the calculation. This greatly reduces the number of words involved in the calculation, making similar words more

| No. | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| 1 | 克/v | 克/v | 克/v | 克/v | 克/v | 克/v | 克/v |
| 2 | 曰/v | 敬/v | **叛/v** | 知/v | 相/v | 十/m | 祭/v |
| 3 | 之/r | **亂/v** | 敬/v | 此/r | 通/v | 設/v | **下/v** |
| 4 | 其/r | 愛/v | **反/v** | 為/v | 祭/v | 謀/v | 辟/v |
| 5 | 有/v | **反/v** | 德/n | **伐/v** | **戰/v** | **戰/v** | **降/v** |
| 6 | 者/r | 謀/v | **亂/v** | 如/v | 秦/ns | 發/v | 聽/v |
| 7 | 為/v | **敗/v** | 愛/v | 是/r | 非/v | 年/n | 中/v |
| 8 | 無/v | **服/v** | 此/r | 人/n | 設/v | 魯/ns | 離/v |
| 9 | 人/n | 故/n | 謀/v | 得/v | 養/v | 神/n | 行/n |
| 10 | 是/r | 待/v | 大/a | 謂/v | 教/v | 合/v | 室/n |
| 11 | 可/v | 德/n | **敗/v** | 曰/v | 重/a | 棄/v | 相/v |
| 12 | 使/v | **勝/v** | **服/v** | 君/n | **敗/v** | 者/n | **逆/v** |
| 13 | 君/n | 善/n | 禮/n | 有/v | 同/a | 通/v | 長/a |
| 14 | 所/r | 罪/n | 是/r | 吾/r | 舍/v | 西/f | 予/r |
| 15 | 能/v | 亡/v | 待/v | 者/r | 棄/v | 刑/n | **止/v** |
| 16 | 我/r | 命/n | 故/n | 之/r | 動/v | 愛/v | 發/v |
| 17 | 得/v | **叛/v** | 善/n | 所/r | 定/v | **下/v** | 晏子/nr |
| 18 | 民/n | 喪/v | 成/v | 何/r | 信/v | 利/v | 將/v |
| 19 | 大/a | 禮/n | 王/n | 其/r | 善/v | 木/n | 敬/v |
| 20 | 此/r | 利/v | 公/n | 用/v | 宋/ns | 正/v | 位/n |

*Table 7.* Word similarity calculation results of the target word "克/v (overcome)".

| Methods / Evaluation | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| **DIRECT** | 29 | 30 | 33 | **39** | 20 | 20 | **24** |
| **DIRECT-AVG** | 1.45 | 1.5 | 1.65 | **1.95** | 1 | 1 | **1.2** |
| **P(1) (%)** | 95 | 95 | 95 | 95 | 95 | 95 | 95 |
| **P(5) (%)** | 19 | 20 | 21 | **22** | 19 | 19 | 19 |
| **P(10) (%)** | 11.5 | 13 | 13 | 13 | 9.5 | 9.5 | 9.5 |
| **INVR** | 1.045 | 1.074 | 1.118 | **1.189** | 0.954 | 0.956 | **0.968** |

*Table 8.* Overall evaluation results of noun similarity

| Methods / Evaluation | CN | Jaccard | Salton | CDSim | LRE | RE | LREW |
|---|---|---|---|---|---|---|---|
| **DIRECT** | 64 | 46 | 54 | **73** | 31 | 29 | 31 |
| **DIRECT-AVG** | 3.2 | 2.3 | 2.7 | **3.65** | 1.55 | 1.45 | 1.55 |
| **P(1)(%)** | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| **P(5) (%)** | 24 | 22 | 23 | **27** | 21 | 21 | 21 |
| **P(10) (%)** | 15.5 | 14 | 15.5 | **21.5** | 11 | 11 | **11.5** |
| **INVR** | 1.347 | 1.219 | 1.309 | **1.468** | 1.048 | 1.038 | **1.074** |

*Table 9.* Overall evaluation results of verb similarity

| No. | Word | S1 | S2 | S3 | S4 | S5 | Count |
|---|---|---|---|---|---|---|---|
| 1 | 百/m | 五十/m | 止/v | 耳/n | 笑/v | 步/n | 5 |
| 2 | 五十/m | 百/m | 止/v | 耳/n | 笑/v | 可/v | 5 |
| 3 | 耳/n | 止/v | 百/m | 五十/m | 笑/v | — | 4 |
| 4 | 王/n | 此/r | 可/v | — | — | — | 2 |
| 5 | 曰/v | 知/v | 用/v | 材木/n | 食/v | 者/r | 7 |
| 6 | 帛/n | 可/v | — | — | — | — | 1 |
| 7 | 穀/n | 用/v | 材木/n | 食/v | 者/r | 曰/v | 6 |
| 8 | 用/v | 材木/n | 食/v | 者/r | 曰/v | 穀/n | 6 |
| 9 | 或/r | 步/n | 笑/v | 者/r | — | — | 3 |
| 10 | 此/r | 王/n | — | — | — | — | 1 |
| 11 | 可/v | 帛/n | 魚/n | 五十/m | 王/n | — | 4 |
| 12 | 者/r | 用/v | 材木/n | 食/v | 或/r | 曰/v | 8 |
| 13 | 知/v | 曰/v | — | — | — | — | 1 |
| 14 | 笑/v | 或/r | 止/v | 耳/n | 步/n | 百/m | 6 |
| 15 | 衣/v | 用/v | 材木/n | 食/v | 者/r | 曰/v | 6 |
| 16 | 魚/n | 可/v | — | — | — | — | 1 |
| 17 | 材木/n | 材木/n | 食/v | 者/r | 曰/v | 穀/n | 6 |
| 18 | 止/v | 耳/n | 百/m | 五十/m | 笑/v | — | 4 |
| 19 | 步/n | 或/r | 笑/v | 者/r | 百/m | — | 4 |
| 20 | 食/v | 材木/n | 食/v | 者/r | 曰/v | 穀/n | 6 |

*Table 10.* CDSim node similarity calculation results

likely to be ranked higher. However, this premise also leads to many words not being considered in the similarity calculation for a given target word, which may miss some potential similar words.

The LREW method calculates the relative entropy of all words in the network, which fundamentally guarantees that no similar words will be missed. Therefore, although the LREW method still has deficiencies in extracting node local information, from the perspective of the comprehensiveness of the calculation results, this method remains a feasible similarity calculation approach. In the next stage, we will further improve the LREW algorithm, try to add the degree and weight information of higher-order neighbor nodes, thereby incorporating more local network information. In addition, we will try to apply the algorithm to more corpora to test its versatility and robustness.

## Acknowledgements

The authors acknowledge the National Social Science Foundation of China (Grant Number: 21&ZD331 and 20ATQ006) for financial support.

## References

Chen, Chih-Ming, and Chung Chang. 2019. "A Chinese ancient book digital humanities research Chen, Chih-Ming, and Chung Chang. 2019. "A Chinese Ancient Book Digital Humanities Research Platform to Support Digital Humanities Research." *The Electronic Library* 37, no. 2: 314-36. https://doi.org/10.1108/EL-10-2018-0213

Chen, Dan, Housheng Su, and Gui-Jun Pan. 2020. "Framework Based on Communicability to Measure the Similarity of Nodes in Complex Networks." *Information Sciences* 524: 241-53. https://doi.org/10.1016/j.ins.2020.03.046.

Curran, James R., and Marc Moens. 2002. "Scaling Context Space." In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, Philadelphia July 2002, pp. 231-23. https://www.aclweb.org/anthology/P02-1030.pdf

Güneş, İsmail, Şule Gündüz-Öğüdücü, and Zehra Çataltepe. 2016. "Link Prediction Using Time Series of Neighborhood-Based Node Similarity Scores." *Data Mining and Knowledge Discovery* 30, no.1:147-80. https://doi.org/10.1007/s10618-015-0407-0

Guo, Shaoru, Yong Guan, Ru Li and Qi Zhang. 2016. "Chinese Word Similarity Computing Based on Combination Strategy." In *Natural Language Understanding and Intel-*

*ligent Applications*, edited by Xue N., Zhao D., Huang X., Feng Y, 744-52. Cham: Springer. doi: https://doi.org/10.1007/978-3-319-50496-4_67

Han, Pu, Dongbo Wang and Hengmin Zhu. 2015. "Research of Chinese Similar Words Mining and Similarity Calculation Based on Complex Network." *Journal of The China Society for Scientific and Technical Information* 34, no. 8: 885-96.

Hu, Yu, Tiezheng Nie, Derong Shen and Yue Kou. 2015. "An Unsupervised Approach for Constructing Word Similarity Network." In *12th Web Information System and Application Conference (WISA)*, IEEE, Jinan, Shangdong, China Sep 11-13, 2015, 265-68. https://ieeexplore.ieee.org/abstract/document/7396648

Huang, Degen, Jiahuan Pei, Cong Zhang, Kaiyu Huang and Jianjun Ma. 2018. "Incorporating Prior Knowledge into Word Embedding for Chinese Word Similarity Measurement." *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17, no.3: 1-21. https://doi.org/10.1145/3182622.

Huang, Shuiqing and Dongbo Wang. 2017. "Review and Trend of Researches on Ancient Chinese Character Information Processing." *Library And Information Service* 61, no. 12: 43-49.

Jiang, Wanchang, and Yinghui Wang. 2020. "Node Similarity Measure in Directed Weighted Complex Network Based on Node Nearest Neighbor Local Network Relative Weighted Entropy." *IEEE Access* 8: 32432-41. https://doi.org/10.1109/ACCESS.2020.2971968.

Konaka, Fumito, and Takao Miura. 2016. "Word Similarity Based on Domain Graph." In *International Conference on Model and Data Engineering*, edited by Bellatreche L., Pastor Ó., Almendros Jiménez J., Aït-Ameur Y, 346-57. Lecture Notes in Computer Science, vol 9893. Cham:Springer. https://doi.org/10.1007/978-3-319-45547-1_27

Li, Bin, Ning Xi, Minxuan Feng, and Xiaohe Chen. 2013. "Corpus-Based Statistics of Pre-Qin Chinese." In *Chinese Lexical Semantics: 13th Workshop, CLSW 2012, Wuhan, China, July 6-8, 2012, Revised Selected Papers 13*. Springer Berlin Heidelberg, 145-153.

Li, Bo, Jiyu Wei,Yang Liu,Yuze Chen,Xi Fang, and Bin Jiang. 2021. "Few-shot Relation Extraction on Ancient Chinese Documents." *Applied Sciences* 11, no.24: 12060. https://doi.org/10.3390/app112412060

Li, Shibao, Junwei Huang, Zhigang Zhang, Jianhang Liu, Tingpei Huang, and Haihua Chen. 2018. "Similarity-Based Future Common Neighbors Model for Link Prediction In Complex Networks." *Scientific Reports* 8: 17014. https://doi.org/10.1038/s41598-018-35423-2

Li, Si, Mingzheng Li, Yajing Xu, Zuyi Bao, Lu Fu and Yan Zhu. 2018. "Capsules Based Chinese Word Segmentation for Ancient Chinese Medical Books." *IEEE Access* 6: 70874-83 https://doi.org/10.1109/ACCESS.2018.2881280

Lü, Linyuan, and Tao Zhou. 2011. "Link Prediction in Complex Networks: A Survey." *Physica A: Statistical Mechanics and Its Applications* 390: 1150-1170. https://doi.org/10.1016/j.physa.2010.11.027

Ma, Xiuxia, Xiangfeng Luo, Subin Huang and Yike Guo. 2019. "Multi-Distribution Characteristics Based Chinese Entity Synonym Extraction from The Web." *International Journal of Intelligent Information Technologies (IJIIT)* 15, no. 3: 42-63. https://doi.org/ 10.4018/IJIIT.2019070103

Pablo, E., and Kyomin Jung. 2016. "Knowledge Extraction Through Etymological Networks: Synonym Discovery in Sino-Korean Words." In *IEEE International Conference on Knowledge Engineering and Applications (ICKEA)*, IEEE, Singapore September 28-30 2016. https://ieeexplore.ieee.org/abstract/document/7803019/

Ren, Xiang, and Tao Cheng. 2015. "Synonym Discovery for Structured Entities on Heterogeneous Graphs." In *Proceedings of the 24th International Conference on World Wide Web*, Florence, Italy, May 18, 2015, 443-53. https://doi.org/10.1145/2740908.2745396

Wang, Hongbin, Haibing Wei, Jianyi Guo, and Liang Cheng. 2019. "Ancient Chinese Sentence Segmentation Based on Bidirectional LSTM+ CRF Model." *Journal of Advanced Computational Intelligence and Intelligent Informatics* 23, no.4: 719-25. https:// 10.20965/jaciii.2019.p0719

Wang, Like, Yuan Sun, and Xiaobing Zhao. 2018a. "English-Chinese Cross Language Word Embedding Similarity Calculation." In *Proceedings of the 2018 Artificial Intelligence and Cloud Computing Conference*, Tokio, Japan, December 21-23 2018, ed. Chiharu Ishii. https://doi.org/10.1145/3299819.3299831

Wang, Ruili, Wanting Ji, and Baoyan Song. 2018b. "Durable Relationship Prediction And Description Using a Large Dynamic Graph." *World Wide Web* 21: 1575-1600. https://doi.org/10.1007/s11280-017-0510-9

Wang, Yonggen, Yanhui Gu, Junsheng Zhou and Weiguang Qu. 2015. "A Graph-Based Approach for Semantic Similar Word Retrieval." In *International Conference on Behavioral, Economic and Socio-cultural Computing (BESC)*, IEEE, Nanjing, China, October 30 – November 1, 2015, 24-27. https://ieeexplore.ieee.org/abstract/document/7365952

Wen, Tao, Shuyu Duan, and Wen Jiang. 2019. "Node Similarity Measuring in Complex Networks With Relative Entropy." *Communications In Nonlinear Science and Numerical Simulation* 78: 104867. https://doi.org/10.1016/j.cnsns.2019.104867

Yang, Bo, Tao Huang, and Xu Li. 2019. "A Time-Series Approach to Measuring Node Similarity in Networks and Its Application to Community Detection." *Physics Letters*

*A* 383, no. 30: 125870. https://doi.org/10.1016/j.physleta.2019.12587

Yang, Yu, Jian Pei, and Abdullah Al-Barakati. 2017. "Measuring in-Network Node Similarity Based on Neighborhoods: A Unified Parametric Approach." *Knowledge and Information Systems* 53, no. 1: 43-70. https://doi.org/10.1007/s10115-017-1033-5

Yin, Fulian, Yanyan Wang, Jianbo Liu, and Meiqi Ji. 2020. "Enhancing Embedding-Based Chinese Word Similarity Evaluation with Concepts and Synonyms Knowledge." *CMES-Computer Modeling in Engineering & Sciences* 124: 747-64. https://doi.org/10.32604/cmes.2020.010579.

Zhang, Qi, Meizhu Li, and Yong Deng. 2018. "Measure the Structure Similarity of Nodes in Complex Networks Based on Relative Entropy." *Physica A: Statistical Mechanics and its Applications* 491: 749-63. https://doi.org/10.1016/j.physa.2017.09.042

Zhu, Tianchen, Zhaohui Peng, Xinghua Wang, and Xiaoguang Hong. 2017. "Measuring the Similarity of Nodes in Signed Social Networks with Positive and Negative Links." In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint Conference on Web and Big Data*, edited by Chen L., Jensen C., Shahabi C., Yang X. and Lian X., 399-407. Cham: Springer. https://doi.org/10.1007/978-3-319-63579-8_31