Mining the rank of universities with the encyclopedia Wikipedia

Zongjian Li1, Cong Li1,2* & Xiang Li1,2

1Adaptive Networks and Control Lab, Department of Electronic Engineering, Fudan University, Shanghai 200433, P.R.China; 2Research Center of Smart Networks and Systems, School of Information Science Engineering, Fudan University, Shanghai 200433, P.R.China

Appendix A Fetching the network of Wikipedia articles

The web crawler fetches the HTML file, which contains the hyperlinks of each Wikipedia article. The hyperlink \( L_{i,j} \) is a directed link, which directs from a Wikipedia article \( i \) to another Wikipedia article \( j \). The obtained Wikipedia data can be mapped into a raw Wikipedia network, where the Wikipedia articles are vertices, and the hyperlink between a pair of Wikipedia articles is the edge (link) between the corresponding two vertices. Note that the article “Main Page” and article “UTC”, which are linked from all articles, are ignored in this work. Algorithm A1 illustrates how the raw data is collected and stored into the database.

The network generated from the raw data contains multiple nodes, since articles may have aliases. For instance, the article titled with “Fudan University” may be corresponding to the node “Fudan” with the URL (en.wikipedia.org/wiki/Fudan) or the node “Fudan University” with URL (en.wikipedia.org/wiki/Fudan University). In this work, we only consider the simple networks, which do not contain multiple nodes. Therefore, we merge all the alias of the same article into one node and redirect all the corresponding URLs to the merged node. The newly generated network is named the Wikipedia article reference network (WARN). Algorithm A2 depicts the procedure of generating WARN with the network database. The process ensures that each node in the network is a unique Wikipedia article.

Appendix B Data Description

The geographic distribution of the 114 universities is shown in Figure B1. The 114 universities are located in 18 countries (or regions). Totally, 72 universities are located in six English-speaking regions. Thirty-eight universities are in the United States, and twenty universities are in the United Kingdom.

* Corresponding author (email: cong_li@fudan.edu.cn)
Algorithm A1 Fetch data using crawler program and network database.

Ensure:

1. Raw network $R = (N, L)$;
2. Redirect information dictionary $D$;
3. $D \leftarrow \emptyset$
4. $N \leftarrow \{\text{MainPage}\}$
5. $L \leftarrow \emptyset$
6. $S \leftarrow \{\text{MainPage}\}$  
   ▷ A set to store unvisited article web pages.
7. while $S \neq \emptyset$ do
8.     $p \leftarrow \text{Pop}(S)$
9.     $a \leftarrow \text{DownloadArticle}(p)$
10.    $t \leftarrow \text{GetArticleTitleInHtml}(a)$
11.    if $t \neq p$ then
12.        $D[p] \leftarrow t$
13.    else
14.        for all $h \in \text{GetHyperLinks}(a)$ do
15.            if IsHyperLinkForArticle($h$) then
16.                $N \leftarrow N \cup \{h\}$
17.                $L \leftarrow L \cup \{(p, h)\}$
18.            end if
19.        end for
20.    end if
21. end while
22. $N \leftarrow N - \{\text{MainPage}, UTC\}$
23. for all $n \in N$ do
24.     $L \leftarrow L - \{(n, \text{MainPage}), (n, UTC)\}$
25. end for
26. return $R, D$
Algorithm A2 Generate WARN from the raw network in network database.

Require:
1: Raw network $R = (N, L)$;
2: Redirect information dictionary $D$;

Ensure: $\text{WARN} = (V, E)$;

3: $V \leftarrow \emptyset$
4: for all $v \in N$ do
5:   if $v \notin \text{GetKeys}(D)$ then
6:     $V \leftarrow V \cup \{v\}$
7:   end if
8: end for
9: $E \leftarrow \emptyset$
10: for all $(v, w) \in L$ do
11:   if $w \in \text{GetKeys}(D)$ then
12:     $u \leftarrow D[w]$ 
13:     $E \leftarrow E \cup \{(v, u)\}$
14:   else
15:     $E \leftarrow E \cup \{(v, w)\}$
16:   end if
17: end for
18: return $W$

Appendix C  Relation between reputation criterions and the QS or THE ranking

The relation between the intuitive reputation criterion, the potential reputation criterion or the network-based criterion and the QS or THE ranking is shown in Figure C2 and Figure C1. The "T" and "t" ("Q" and "q") marks denote that their x-axis is the THE (QS) ranking. Marks in upper-case letter denote universities located in an English-speaking region, and the lower-case letter marks represent the universities in a non-English-speaking regions.

Figure C1  Relation between the in-degree and out-degree of university articles in Wikipedia and QS or THE ranking.
Appendix D  Algorithms for EWARS generator

We propose to generate a pre-EW ARS, which includes not only the targeted articles but also the articles related to the targeted articles, as the preparation for the EW ARS. Then, we perform the path length method or the local vertex connectivity method (Algorithms are introduced below) on a pre-EW ARS to obtain the EW ARS.

Appendix D.1 Pre-EWARS

Two parameters, i.e. the direction parameter and the depth parameter should be set for generating the pre-EW ARS. The direction parameter is regarded as a filter through which only a specific kind of directed links pass into the pre-EW ARS from the WARN. The depth parameter is the recursive depth that is used to limit the data mining depth. Algorithm D1 formulates a procedure of creating a pre-EW ARS. In the initial step, we set the pre-EW ARS as an empty network. Then, we find the nearest neighbors along the in-coming or out-going links from the university articles and add the links and articles to the pre-EW ARS. Next, we try to find the neighbors of the nearest neighbors if the depth parameter is larger than 1. All articles of the pre-EW ARS will be found by recursively executing the “find and add” operation, until the depth limit is satisfied. Finally, we obtain an undirected pre-EW ARS by adding inverse links (see Algorithm D1).

Appendix D.2 Path length method for generating EWARS

In the path length (PL) method, we use the shortest path length to represent the relation weight between two university articles. First, we generate a pre-EW ARS. Second, the shortest path lengths between any two university articles are calculated. Third, we set the shortest path lengths as the relation weights between universities in the EW ARS. Algorithm D2 shows the process of generating the EW ARS.

Appendix D.3 Local vertex connectivity method for generating EWARS

Traditionally, the vertex connectivity $\nu$ of a network $G$ is defined as the minimum number of nodes whose removal disconnects the network. A network with $\nu > 0$ is connected, and a network with $\nu = k$ is said to be $k$-connected. In this work, we propose to utilize a local vertex connectivity method to generate EWARS. Note that the local vertex connectivity of two directly connected nodes does not exist, since we cannot disconnect two directly connected nodes by removing other nodes. Hence, when we calculate the $\nu_{ij}$, we do not take the direct connection $L_{ij}$
Algorithm D1 Generate a pre-EWARS from given WARN.

Require:
- WARN $W$
- Universities $U$
- Edge direction $d$
- Recursive depth $r$

Ensure: Pre-EWARS $P = (N, L)$

\[
N \leftarrow \emptyset \\
L \leftarrow \emptyset \\
\text{for all } u \in U \text{ do} \\
\text{AddVertex}(u, r) \\
\text{end for} \\
\text{return } P \\
\]

procedure AddVertex($v$, $r$)

if $r > 0$ then

if $d = OUT$ then

for all $(v, w) \in W$ do

$N \leftarrow N \cup \{w\}$

$L \leftarrow L \cup \{(v, w), (w, v)\}$

AddVertex($w$, $r - 1$)

end for

else

for all $(w, v) \in W$ do

$N \leftarrow N \cup \{w\}$

$L \leftarrow L \cup \{(v, w), (w, v)\}$

AddVertex($w$, $r - 1$)

end for

end if

end if

end procedure
Algorithm D2 Generate a EW ARS from given pre-EW ARS using PL method.

**Require:**
- Pre-EW ARS $P$;
- Universities $U$;

**Ensure:** EW ARS $G = (V, E)$;

\[
\begin{align*}
V & \leftarrow \emptyset \\
\text{for all } u \in U & \text{ do} \\
V & \leftarrow V \cup \{u\} \\
\text{end for} \\
E & \leftarrow \emptyset \\
\text{for all } u \in U & \text{ do} \\
\text{for all } v \in U & \text{ do} \\
\text{if } (u, v) \notin E \text{ and } (v, u) \notin E & \text{ then} \\
w & \leftarrow \text{ShortestPathLength}(P, u, v) \\
E & \leftarrow E \cup \{(u, v, w), (v, u, w)\} \\
\text{end if} \\
\text{end for} \\
\text{end for} \\
\text{return } G = (V, E)
\end{align*}
\]

between two vertices $i$ and $j$ into consideration (see Figure D1). The local vertex connectivity (LVC) method works similarly to the PL method, that we first generate a pre-EW ARS. Then, the local vertex connectivity between any two university articles are calculated as weights of the edges between universities in the EW ARS. The procedure of the LVC method is described in Algorithm D3.

**Figure D1** An example of the local vertex connectivity $\nu_{12} = 1$ for two directly connected nodes, university 1 and university 2.
Algorithm D3 Generate a EW ARS from given pre-EW ARS using LVC method.

Require:
- Pre-EWARS $P = (N, L)$;
- Universities $U$;

Ensure: EWARS $G = (V, E)$;

\[
\begin{align*}
V & \leftarrow \emptyset \\
\text{for all } u \in U & \text{ do} \\
& \quad V \leftarrow V \cup \{u\} \\
\text{end for} \\
E & \leftarrow \emptyset \\
\text{for all } u \in U & \text{ do} \\
& \quad \text{for all } v \in U \text{ do} \\
& \quad \quad \text{if } (u, v) \notin E \text{ and } (v, u) \notin E \text{ then} \\
& \quad \quad \quad T \leftarrow L \\
& \quad \quad \quad L \leftarrow L - \{(u, v), (v, u)\} \\
& \quad \quad \quad w \leftarrow \text{VERTEXCONNECTIVITY}(P, u, v) \\
& \quad \quad \quad \text{Calculate edge weight} \\
& \quad \quad \quad L \leftarrow T \\
& \quad \quad \quad E \leftarrow E \cup \{(u, v, w), (v, u, w)\} \\
& \quad \quad \text{end if} \\
& \quad \text{end for} \\
& \text{return } G = (V, E)
\end{align*}
\]