epl A Letters Journal Exploring the Frontiers of Physics

July 2008

EPL, **83** (2008) 18002 www.epljournal.org
doi: 10.1209/0295-5075/83/18002

What role does syntax play in a language network?

Haitao Liu[a] and Fengguo Hu

Institute of Applied Linguistics, Communication University of China - CHN-100024 Beijing, China

received 17 April 2008; accepted in final form 19 May 2008
published online 13 June 2008

PACS 89.75.Hc – Networks and genealogical trees
PACS 89.90.+n – Other topics in areas of applied and interdisciplinary physics

Abstract – That almost all language networks are small-world and scale-free raises the question of whether syntax plays a role to measure the complexity of a language network. To answer this question, we built up two random language (dependency) networks based on a dependency syntactic network and investigated the complexity of these three language networks to see if the non-syntactic ones have network indicators similar to the syntactic one. The results show that all the three networks are small-world and scale-free. While syntax influences the indicators of a complex network, scale-free is only a necessary but not sufficient condition to judge whether a network is syntactic or non-syntactic. The network analysis focuses on the global organization of a language, it may not reflect the subtle syntactic differences of the sentence structure.

Introduction. – Previous studies [1–5] investigating into language networks, which are built on different principles, show that they are small-world and scale-free, just like most other real-world networks [6]. Questions remain, however, if all language networks have properties such as small-world and scale-free: Could they be viewed as a general feature of a language network? What role does syntax play in such a syntactic (language) network? If dependencies are built by randomly linking words in the same sentence, would the network still follow the properties similar to the syntactic one? Can the local (micro) syntactic analysis in a sentence be reflected in the global (macro) properties of a language network?

The present paper intends to answer these questions. To do that, syntactic networks are chosen to be the object of study, because "[T]he vast expressive power of human language would be impossible without syntax, and the transition from non-syntactic to syntactic communication was an essential step in the evolution of human language. (see [7], p. 495).

Presented in the second section are some formal fundamentals on dependency syntax and means to generate a random dependency graph based on a syntactic dependency graph. Results of the network analysis of one syntactic network and two corresponding random networks are discussed in the third section, followed by concluding remarks and directions for further research in the fourth section.

[a] E-mail: lhtcuc@gmail.com

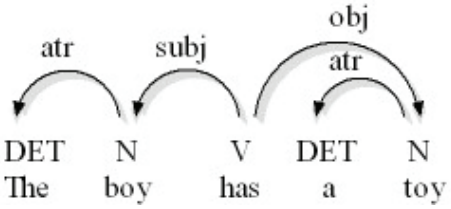

Fig. 1: Syntactic dependency structure of *The boy has a toy.*

Formal description of the dependency analysis. – Dependency approaches have many advantages, as discussed in detail in [8]. For the present study, a dependency approach was employed according to which the syntactic structure of a sentence consists of nothing but the dependencies between individual words. Figure 1 shows a dependency analysis of *The boy has a toy.*

In fig. 1, all the words in a sentence are connected together by grammatical relations. For example, the subject and object depend on the main verb; the determiner depends on the nouns that they modify; and so on.

A formal expression can be given to describe how the dependency analysis generates random dependency corpora:

Given a sentence S with the length of $n\,(n>1)$,

$$S=(x_1,x_2,\ldots,x_n),$$

where $x_i\,(1\leqslant i\leqslant n)$ is the i-th word in S and i is accordingly called the word order of x_i in S.

After parsing a sentence based on the dependency syntax, a dependency graph can be obtained. The dependency graph of the sentence S often includes three elements: the words, the POS (part of speech) of the words and the dependency relations between two words (governor and dependent). The first two elements can be considered as functions of the word order i, and the third as a function of ordered pairs of word orders.

A dependency grammar in language L consists of a word list, a POS list and the dependency relation list. For parsing a sentence S, the words are put into a one-to-one correspondence with their word orders, so the words can be recognized as a function of word orders:

$$x_i = f_{word}(i), \quad 1 \leqslant i \leqslant n.$$

$(pos_1, pos_2, \ldots, pos_n)$ is the sequence of POSs of this sentence and it can be described as a function of word orders:

$$pos_i = f_{tag}(i), \quad 1 \leqslant i \leqslant n.$$

Dependency parsing links all words in S using dependency relations, which can be expressed by a number of ordered triples as follows:

$$\langle i, j, rname_{ij} \rangle.$$

Here i is the word order of the governor, j is the word order of the dependent, and $rname_{ij}$ is the type of the dependency relation between x_i and x_j. Thus, the dependency relation type can be defined as a function of ordered pairs $\langle i, j \rangle$:

$$rname_{ij} = f_{relation}(\langle i, j \rangle),\ 1 \leqslant i, j \leqslant n \text{ and } x_i \text{ governs } x_j.$$

In this way, a dependency graph D of S is defined as

$$\begin{aligned}
&D = (V, E, f_{word}, f_{tag}, f_{relation}),\\
&V = \{1, 2, \ldots, n\},\\
&E \in V^2,\\
&f_{word}\colon V \to W,\\
&f_{tag}\colon V \to T,\\
&f_{relation}\colon E \to R,\\
&W = \{w_1, w_2, \ldots, w_{MAXW}\}, \quad MAXW \geqslant 1,\\
&T = \{t_1, t_2, \ldots, t_{MAXT}\}, \quad MAXT \geqslant 1,\\
&R = \{r_1, r_2, \ldots, r_{MAXR}\}, \quad MAXR \geqslant 1,
\end{aligned}$$

where W is the word list, T is the POS list, and R is the dependency relation list in L.

D is a well-formed dependency graph if and only if E satisfies the following four conditions [9,10]:

1) Single-governor:

$$(\forall x \in V)(\langle x, x \rangle \notin E \wedge ((\exists y, z \in V)(\langle y, x \rangle \in E \wedge \langle z, x \rangle \in E)) \Rightarrow (y = z)).$$

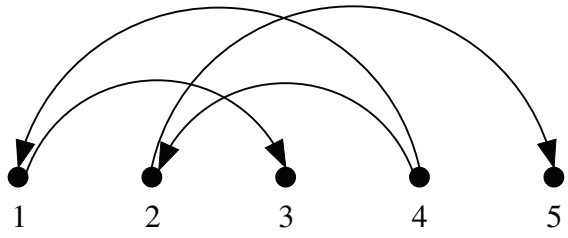

Fig. 2: A dependency graph with crossing arcs.

In a dependency graph, a word has at most a governor, which should not be equal to itself.

2) Single-root:

$$(\exists x \in V)((\forall y \in V)(\langle y, x \rangle \notin E) \wedge ((\exists z \in V)(\forall w \in V)(\langle w, z \rangle \notin E) \Rightarrow (z = x))).$$

The dependency graph of a sentence has one and only one root. As a default rule we call the unique element as *root* of a dependency graph.

3) Connectedness:

$$\begin{aligned}
&\forall (x \in V)(\exists v_1, v_2, \ldots, v_k \in V)(\langle root, v_1 \rangle \in\\
&E \wedge \langle v_1, v_2 \rangle \in E \wedge \cdots \langle v_k, x \rangle \in E)), \quad 0 \leqslant k \leqslant n-2.
\end{aligned}$$

There are directed paths from the vertex labeled *root* to all other vertices. $x \xrightarrow{*} y$ shows a directed path from vertex x to vertex y. These three conditions imply that a well-formed dependency graph is also acyclic:

$$(\forall x \in V)(x \xrightarrow{*} x).$$

Satisfying conditions (1)–(3) cannot guarantee that there are no crossing arcs, as shown in fig. 2.

There are dependency graphs with crossing arcs in some languages. However, no-crossing arcs are often considered as a condition of a well-formed dependency graph for constructing a more efficient parsing algorithm. To form a graph without crossing arcs, the serial numbers of all vertices which can be reached from vertex x should be continuous —this is called *continuity* condition of a graph. *Continuity*, also called *projectivity*, was first discussed in [11,12], and then given a formal definition in [10].

A new notion of *reachable domain* in the present paper is introduced to describe continuity. The reachable domain A_x of vertex x is a set of vertices that are reachable from x and x itself:

$$A_x = \{x\} \cup \{y | x \xrightarrow{*} y\}, \quad x \in V.$$

Using A_x, we can define *continuity* as

4) Continuity:

$$(\forall x \in V)(|A_x| = \max(A_x) - \min(A_x) + 1),$$

where $|A_x|$ is the number of elements of set A_x. $\max(A_x)$ denotes the maximum value of A_x and $\min(A_x)$ the

minimum value of A_x. For example, while fig. 1 is a well-formed dependency graph, the vertices 1 and 2 in fig. 2 violate the *continuity* condition. So, the graph contains crossing arcs.

Theoretically, a language could be produced with a randomly generated lexicon and sentences, but it is impossible to syntactically analyze such a language or build a syntactic network for the sentences. Thus, we will randomly assign the governor for every word in a sentence to generate a random dependency analysis of a sentence based on the original syntactic analysis.

To describe the syntactic well formedness of the analysis of a sentence, we randomly generated two dependency graphs satisfying conditions (1)–(2) and conditions (1)–(4), respectively. We removed all dependency relations in a well-formed dependency graph and randomly assigned the governor for all words in a sentence according to different constraints.

Supposing that there are $n\,(n \geqslant 1)$ words in a sentence with the word orders $1, 2, \ldots, n$, the algorithm attempts to assign a number to each word as its governor's word order. The *root* of the dependency graph has no governor, so the algorithm will assign 0 to it as its governor's order number. The two algorithms are stated as follows.

Algorithm 1: Randomly generating dependency graphs satisfying conditions (1) and (2). Firstly, choose a number from 1 to n randomly as root and assign 0 to the root as its governor. Then, assign one randomly generated number between 1 and n to each remaining word as its governor but which is not the word itself. Figure 2 shows one of such dependency graphs.

In the random graph generated by algorithm 1 (hereinafter as RL1), we select one word as root within each sentence, disregarding syntax and meaning; and then, for each of the remaining words, we randomly selected a word in the same sentence as its governor.

Algorithm 2: Randomly generating dependency structure graphs satisfying conditions (1)–(4). This algorithm builds dependency graphs increasingly like a snowball. Firstly, generate a number from 1 to n randomly as root and assign 0 to the root as its governor. Then, using the *root* as the start point of the dependency graph, select a word randomly from the words which still have no governor, randomly assign a governor for it from the graph, attach the word into the graph. During this process, the *continuity* principle should not be violated. Repeat the process until all words are added into the dependency graph. This algorithm can generate a dependency graph such as that in fig. 3.

In the random graph generated by algorithm 2 (hereinafter RL2), only dependency trees are generated satisfying the conditions (1)–(4), *i.e.* without crossing arcs, while the governor is assigned to a word.

Three dependency graphs for a sentence can be made following the methods mentioned above. The first as in fig. 1 is syntactic, RL1 as in fig. 2 has the lowest syntactic degree, and RL2 in fig. 3 is more syntactic than RL1.

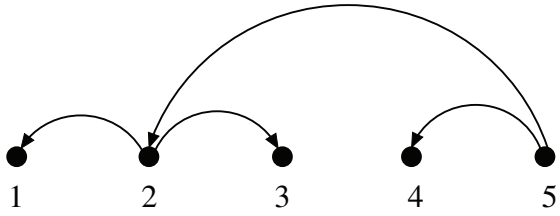

Fig. 3: Random dependency graph without crossing arcs.

In a syntactic network, a vertex is a word (type), and the edge is the relation between two words. Researchers use the word (type) as vertex of a syntactic network in a common way, but they prefer to use various means to build links between two words. From these ways, dependency syntax describes a sentence based on an asymmetrical binary relation between two words that makes it into a more natural choice for building syntactic (language) networks of human language [3,5,13].

The author of ref. [5] proposes a method that converts a dependency syntactic treebank into a syntactic network. A treebank is a collection of dependency graphs for many sentences. Treebanks are often used in computational linguistics [14]. In this paper, we used a dependency treebank that was built on the news (*xinwen lianbo*, hereinafter as xwlb) of China Central Television, a genre which is similar to the written reports. The treebank includes 16654 word tokens. Based on this treebank, we have built two random dependency treebanks with the same words, but with random-generated governors. Following the methods proposed in [5], three undirected networks were built for further investigation. The results and discussion are presented in the section which follows.

Analysis of three dependency networks. – The average path length, clustering coefficients and the degree distribution of a network are among the most frequently investigated network indicators for evaluating the complexity of a network.

The average path length $\langle d \rangle$ is defined as the average shortest distance between any pair of vertices in a network:

$$\langle d \rangle = \frac{2}{N(N+1)} \sum_{i \geqslant j} d_{ij}. \tag{1}$$

Here, N is the number of vertices in the network; d_{ij} is the distance between the vertices i and j, which can be defined as the number of edges in the shortest path linking the two vertices. The three networks in the present study have the same N (4015), but with different $\langle d \rangle$. The syntactic one has the greatest value of 3.372, the $\langle d \rangle$ of RL1 and RL2 is closer, 3.147 and 3.129, respectively.

The diameter D is defined as the longest shortest path in a network. For instance, in the syntactic network, such path is found from the vertex 821 to the vertex 3032, because there are 10 edges in this path, thus, the diameter of the network is 10. RL1 and RL2 have the same D (9).

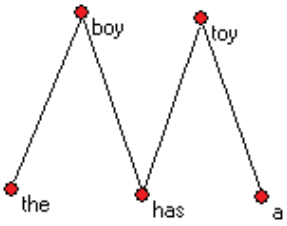

Fig. 4: Network structure of *The boy has a toy.*

In a syntactic network, the number of links of a given word type is called its degree k, which works in a very similar manner as valency in dependency and valency grammar [5]. $\langle k \rangle$ is the average degree of a network. From the three networks, the syntactic $\langle k \rangle$ (6.48) is less than RL1 (7.80) and RL2 (7.95). The degree distributions are defined as the frequency $P(k)$ of having a word type with k links.

The clustering coefficient is the probability that two vertices (*e.g.*, word types) that are neighbors of a given vertex are neighbors of each other. This can be better explained with fig. 4, which is the network's structure of the sentence in fig. 1.

For the syntactic network in fig. 4, the clustering coefficient reflects the probability that two words has a link, as between *the* and *has*, *has* and *a*, *boy* and *toy*.

Let k_i denote the degree of vertex i, E_i denotes the number of edges among the vertices in the nearest neighborhood of vertex i. Then, the clustering coefficient C_i of the vertex i is defined as

$$C_i = \frac{2E_i}{k_i(k_i - 1)}. \tag{2}$$

The clustering coefficient of the network is given by the average of C_i over all the vertices in the network:

$$C = \frac{1}{N} \sum_{i=1}^{N} C_i. \tag{3}$$

Figure 4 shows that there is a less probable link between *a* and *has* in the syntactic network, while two random networks allow to make such link. This is confirmed by the data of the present study. From three networks, the clustering coefficient of the syntactic network is 0.128; RL1, 0.185; RL2, 0.175. In other words, the clustering coefficient is increasing with the randomness of the network.

If a network has a high clustering coefficient C and a very short path length $\langle d \rangle$, it is a small-world (SW) network [15]. In other words, a small-world graph can be defined as a network such that $\langle d \rangle \sim \langle d_{rand} \rangle$ and $C \gg C_{rand}$.

If the degree distribution of a network is following a power law,

$$P(k) \sim k^{-\gamma}. \tag{4}$$

The network is a scale-free network, if the constant γ is between 2 and 3 [16]. In (4), $P(k)$ is the fraction of vertices

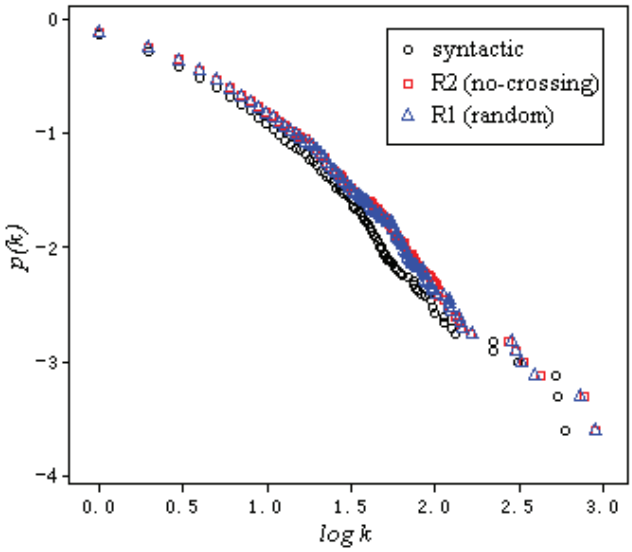

Fig. 5: Cumulative degree distributions of the three networks. Their cumulative degree distributions were fitted by a power law with slopes −1.401 (syntactic), −1.366 (no-crossing) and −1.372 (random), which corresponds to the exponent γ = 2.401, 2.366 and 2.372, respectively.

in the network that have degree k. In other words, $P(k)$ is the probability that a vertex chosen uniformly at random has degree k.

We can calculate the C and $\langle d \rangle$ of the *E-R* random network with the same parameters, whose values are 0.00135 and 4.66.

The above data show that the three networks have average path length similar to *E-R* random networks, but their clustering coefficients are much greater than those of *E-R* random networks. Therefore, all the three dependency networks are small-world networks. Since the degree distributions of the three networks are power law like as fig. 5 reveals, they are also scale-free networks.

Based on the indicators of these networks and *E-R* random networks, we therefore conclude that all the three networks are small-world and scale-free. The data also show that there are some differences between syntactic and non-syntactic networks. Two random dependency networks have parameters closer than those of the syntactic dependency network; however, in spite of the difference between random and syntactic networks, all the networks belong to the same basic network types, so the classification as small-world and scale-free does not distinguish true syntactic networks from random ones.

Two questions remain to be asked: Why do the three networks have the indicators which are reasonable in order to classify them in small-world and scale-free networks? Why is the difference between syntactic network and non-syntactic one greater than that between two non-syntactic networks?

The Zipf law function [17], which is the same for the three networks, probably explains why the three networks

have similar values of network indicators. The authors of ref. [18] argue that in a language network the degree of a word is equivalent to its frequency, whose distribution obeys the Zipf law or the power law. The author of ref. [5] compares two syntactic networks in detail and finds that the degree distributions of syntactic networks are not equivalent to their Zipf curves in the strict sense. In this way, the Zipf law maybe is helpful to explain the similarity of the indicators of the three networks to some extent, but it could not be useful to explain the differences between syntactic and random networks, because random texts often follow the Zip law [19].

Syntax tells us that the word plays an important role in human language [8]. If we pay no attention to the agency of the vertex (word) in a network, it is difficult to find the factors that make the difference between syntactic and non-syntactic networks. For instance, in a syntactic network, a vertex (word) may not freely link to other vertices, but the limit is looser in RL1 and RL2. The result, that the average degree $\langle k \rangle$ of the syntactic network is less than that of RL1 and RL2, addresses the question. Figure 4 shows that it is less probable to build a link between the vertices (word) *boy* and *toy* in a syntactic network than in RL1 and RL2 and that it is reasonable to explain why the clustering coefficient of the syntactic network is less than that of RL1 and RL2. The data show that syntax may influence the indicators of a complex network, but it is impossible to explain the role of syntax only based on the standpoint whether a network is small-world or scale-free.

The findings of the present study are helpful to explain why language networks based on different building principles are often small-world and scale-free, because such network indicators do not only reflect the syntactic feature of a language. The author of ref. [20] shows that syntactic links tend to no-crossing, the similarity of RL1 and RL2 in our study demonstrates that such difference cannot be observed through the indicators of complex networks. If non-syntactic and syntactic networks are scale-free, perhaps we might not argue that syntactic rules are just a by-product of scale-free networks [21]. Our findings probably are not enough to dismiss the claim in [21], but they may show that the indicators of complex networks are not enough to study the syntax of human language. Scale-free is only a feature of syntactic networks, it may not be used as a sole means to assess if a network is syntactic or non-syntactic.

Concluding remarks. – All language networks, if built on certain principles, are small-world and scale-free. But other real-world networks have similar features, too. This makes it difficult to claim that small-world and scale-free are basic properties of human languages and the mentioned indicators of complex networks can be used to explore the structure of human languages. Given that syntax is the most important property of human language, we investigated the complexity of a syntactic network and two randomly generated networks based on the syntactic one. Our results show that all the three networks are small-world and scale-free. The syntactic network has lower average degree and clustering coefficient than the two random networks; however, the differences are too small to classify them into two different types of complex network.

Our study also shows that while the network analysis focuses on the global organization of a language, it may not reflect the subtle syntactic differences of the sentence structure. If we disregard the agency of the vertex (word) in a language network, it is difficult to study micro syntactic problems by macro means as a complex network.

Further planned studies include: using a greater treebank to dig the role of syntax in a complex network; building a closer link between a complex network and syntax about the word; combining other micro quantitative findings as in [22] with network analysis. We hope that these planned studies are useful for finding more fitting networks indicators to distinguish syntactic networks from non-syntactic networks and to explore the role of syntax in a language network.

* * *

We thank the referees for insightful comments, Yue Ming and Li Minglin for improving our English, Zhao Yiyi for annotating treebank.

REFERENCES

[1] Ferrer i Cancho R. and Solé R. V., *Proc. R. Soc. London, Ser. B*, **268** (2001) 2261.
[2] Solé R., Corominas B., Valverde S. and Steels L., *Language Networks: Their Structure, Function and Evolution,* Santa Fe Institute Working Paper (05-12-042) 2005.
[3] Ferrer i Cancho R., Solé R. V. and Köhler R., *Phys. Rev. E*, **69** (2004) 051915.
[4] Li J. and Zhou J., *Physica A*, **380** (2007) 629.
[5] Liu H., *Physica A*, **387** (2008) 3048.
[6] Newman M. E. J., *SIAM Rev.*, **45** (2003) 167.
[7] Nowak M. A., Plotkin J. B. and Jansen V. A., *Nature*, **404** (2000) 495.
[8] Hudson R., *Language Networks: The New Word Grammar* (Oxford University Press, Oxford) 2007.
[9] Mel'čuk I. A., *Dependency Syntax: Theory and Practice* (State University Press of New York, Albany) 1988.
[10] Nivre J., *Inductive Dependency Parsing* (Springer Verlag, Dordrecht) 2006.
[11] Lecerf Y., *Programme des conflits-modèle desconflits, Rapport CETIS*, No. 4, Euratom. (1960) pp. 1–24.
[12] Hays D. G., *Language*, **40** (1964) 511.
[13] Corominas-Murtra B., Valverde S. and Solé R. V., *Emergence of Scale-Free Syntax Networks*, `http://arxiv.org/abs/0709.4344`, 2007.
[14] Abeillé A. (Editor), *Treebank: Building and Using Parsed Corpora* (Kluwer, Dordrecht) 2003.

[15] Watts D. J. and Strogatz S. H., *Nature*, **393** (1998) 440.
[16] Barabási A.-L. and Albert R., *Science*, **286** (1999) 509.
[17] Zipf G. K., *Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology* (Addison-Wesley Press, Cambridge, Mass.) 1949.
[18] Masucci A. P. and Rodgers G. J., *Phys. Rev. E*, **74** (2006) 026102.
[19] Li W., *IEEE Trans. Inf. Theory*, **38** (1992) 1842.
[20] Ferrer I Cancho R., *Europhys. Lett.*, **76** (2006) 1228.
[21] Solé R., *Nature*, **434** (2005) 289.
[22] Liu H., *Glottometrics*, **15** (2007) 1.