Physica A 496 (2018) 620–634

Contents lists available at ScienceDirect

Physica A

journal homepage: www.elsevier.com/locate/physa

Multi-language naming game

Jianfeng Zhou, Yang Lou, Guanrong Chen, Wallace K.S. Tang *

Department of Electronic Engineering, City University of Hong Kong, Hong Kong Special Administrative Region

HIGHLIGHTS

- A new naming game model simulating the multi-language environment is proposed.
- Simulations on the new game are performed on several typical network topologies.
- The probability of each communication pattern can be theoretically estimated.
- A power-law-like relationship between two indices is observed.
- The new game has potential applications in understanding multi-languages emerge and evolution.

ARTICLE INFO

Article history:
Received 11 July 2017
Received in revised form 10 November 2017
Available online 8 January 2018

Keywords:
Naming game
Multi-language
Complex network
Agreement process

ABSTRACT

Naming game is a simulation-based experiment used to study the evolution of languages. The conventional naming game focuses on a single language. In this paper, a novel naming game model named multi-language naming game (MLNG) is proposed, where the agents are different-language speakers who cannot communicate with each other without a translator (interpreter) in between. The MLNG model is general, capable of managing k different languages with $k \geq 2$. For illustration, the paper only discusses the MLNG with two different languages, and studies five representative network topologies, namely random-graph, WS small-world, NW small-world, scale-free, and random-triangle topologies. Simulation and analysis results both show that: 1) using the network features and based on the proportion of translators the probability of establishing a conversation between two or three agents can be theoretically estimated; 2) the relationship between the convergence speed and the proportion of translators has a power-law-like relation; 3) different agents require different memory sizes, thus a local memory allocation rule is recommended for saving memory resources. The new model and new findings should be useful for further studies of naming games and for better understanding of languages evolution from a dynamical network perspective.

1. Introduction

Naming game (NG) [1] studies the emergence of shared lexicons in a population of agents about some objects they observed. It is a kind of simulation-based experiment using computers as the tool. The simplest version of NG is called the minimal naming game [2], in which the agents reach a state of consent after some actions and interactions through iterations. Behavioral *action* means getting a word from the agent's memory, the external lexicons, or inventing a word if there is no corresponding item in the memory; while *interaction* means the transmission of words among the agents. Finally, the process is terminated if a global consensus has been achieved. By experimenting the game with different parameters on

* Corresponding author.
E-mail address: eekstang@cityu.edu.hk (W.K.S. Tang).

https://doi.org/10.1016/j.physa.2017.12.124
0378-4371/

different topologies of networks subject to different rules, it is possible to explain some phenomena emerging from linguistic activities and even some features and principles of linguistic conventions.

The rule of a minimal NG model [3] is as follows: initial every agent with an empty memory, which can be finite [4] or infinite [5–8]. In each iteration, randomly pick two agents, a speaker and a hearer, from the population who are neighbors to each other on the communication network. Then, the speaker says a word to the hearer. The word is randomly picked from the speaker's memory or, if it has an empty memory, from external lexicons. If, by coincidence, the hearer has the same word in its memory, which means a *success* conversation, namely, they reach a consensus, then both of them will clear their memories except keeping the common word; otherwise, this iteration is a *failure* and the hearer will add the new word into its own memory. Running this process iteratively until the whole population reaches the global consensus for which all the agents have one and only one common word in their memories.

Several novel models based on the minimal NG model have been proposed. Baronchelli [5] studied the feedback and broadcasting mechanisms in NG and divided the *interactions* into two schemes: hearer-only NG (HO–NG) and speaker-only NG (SO–NG). Only hearers will update their memories in HO–NG, while only speakers will update their memories in SO–NG. It is found in [5] that the SO–NG is significantly slower regarding the convergence speed. Thus, a multi-hearer scheme was formulated in [5], in which all the neighbors of the speaker would be hearers receiving the same word from the speaker simultaneously.

Li et al. [6] proposed a model, called NGMH (Naming Game with Multiple Hearers), and studied the relationship between the convergence time and the number of hearers. It is shown in [6] that if the number of hearers is smaller than the average node degree, it will accelerate the speed of convergence when adding more hearers; otherwise, the situation will be opposite on scale-free (heterogeneous) networks. In [7], a model with more hearers is also considered but the hearers are randomly chosen from the network. It indicates that with more hearers, the agents reach a faster agreement while the requirement of memory size is significantly reduced.

Considering a real-life scenario where people sometimes make mistakes when talking with each other, Lou and Chen [8] proposed an NG model with learning error (NGLE). Agents in NGLE may transmit a wrong word, with a certain probability. The hearers do not know whether the information received is correct or not, but on the other hand the speaker will gradually reduce the probability of making mistakes as its experience being a speaker increases. It is found in [8] that the communication process with learning errors will increase the requirement of memory and slightly reduce the convergence speed. Considering the existence of zealots in real-life, Gun et al. proposed a model with zealots in [9] and studied the conditions under which one would win.

In [10], NG is studied on multi-local world networks, and in [11] a multi-word NG model is proposed, where agents communicate with each other by sentences, not by single word. In [12], network with adaptive connecting weights is considered, where the weights are depended on the success rate of past communications. Moreover, the case when memory loss happens to some agents is discussed in [13]. In [14], an extended naming game with biased assimilation (NGBA) is considered. The hearer accepts a received word with a predefined probability even the word is not in one's memory. The work in [14] also reveals that by rewiring the links in the network, the convergence process can be greatly accelerated. Another probability controlled model is proposed in [15], where the threshold of reaching consensus has been studied.

More recently, the multi-language phases of NG are investigated in [16] and [17], which can be useful to detect communities in a given network. A multi-language state of an NG refers to a metastable state when different communities reach local consensus on different names, as observed in the processes of naming game on community-based networks which has also been discussed in [12,17,18].

In particular, in [19] a naming game model called ABC model is proposed, in which agents are initially assigned as different language speakers, aiming to study the process of inventing new species of a language from existing languages. It is worth mentioning that, in the ABC model, agents speaking different languages are assumed to be able to communicate with each other unimpededly without translators.

However, in real life, different-language-speaking people cannot communicate with each other directly in general. This also happens to smart devices, e.g. the Bluetooth of an iOS device cannot connect with that of an Android device even within their accessible distances. In such situations, if these people or devices can find translators to help, then they can communicate successfully. The translators can be those agents who speak more than one language, or some translation software, or even just a dictionary used in conversations.

Within the context of multi-languages, when studying the agreement process of a self-organized system, features like communication patterns, convergence speed, memory size of each agent, and the influence of the proportion of agents in different language groups, etc. are different from those in a single-language environment, but all these are not clearly understood today. Therefore, it is of great importance to have a proper model that can be used to simulate the ubiquitous multi-language phenomenon and the corresponding agreement processes.

Motivated by the above observations, this paper proposes a multi-language naming game model (MLNG) in which agents are labeled as different-language speakers. Differing from the ABC model [19], agents speak different languages cannot communicate with each other unless they have a common translator. The MLNG model is a general model, which can be applied to a many-language scenario. For illustration, only the two-language case with Chinese and English is discussed in the paper.

The contributions of this paper include: (1) by considering more realistic situations, a new naming game model named MLNG that can simulate the agreement process in a multi-language environment is established and compared with some

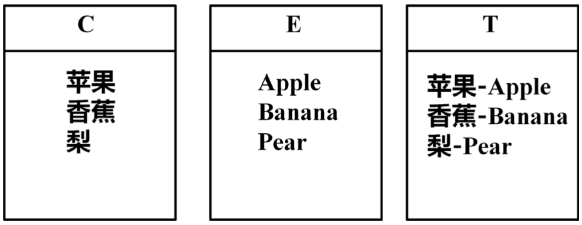


Fig. 1. Agents with different labels: C denotes a Chinese speaker, E denotes an English speaker, while T denotes the translator who can speak both Chinese and English. Both E and C have their own vocabulary while the translator T has both Chinese and English vocabularies with their translations.

existing NG models, revealing its important and unique features; (2) simulations on the MLNG, applied to networks with different topologies, are performed leading to some new findings with in-depth analysis, useful for better understanding of language evolutions.

The rest of the paper is organized as follows. Section 2 introduces the MLNG model. Section 3 shows the simulation results with analysis and comparison. Section 4 concludes the investigation.

2. Multi-Language Naming Game (MLNG)

The MLNG model includes two parts: the agent-based network and the gaming rule. In order to simulate the multi-language context, agents in a network are regarded as different language speakers. When playing the game, in each iteration, one randomly picked agent will make an attempt to transmit a word to one of his neighbors through a pairwise conversation or a three-agent talk. According to the rule of *success* and *failure* introduced below, and iteration by iteration, the game will converge to a state of consent in which all agents share one and only one same word. In order to introduce the MLNG in a more comprehensive way, we firstly explain some important definitions and operations.

Agent assignment: agents in the MLNG are labeled as different language speakers. An example is given in Fig. 1. C represents Chinese speakers. E represents English speakers. And T represents agents who speak both Chinese and English.

Pair-communication: the direct conversation between two agents within the same language group. For example, *pair-communication* could be the direct conversation between C and C, between E and E, and between T and any other language speakers.

Triangle-communication: the conversation between two agents who speak different single-languages with the help of a translator. For example, the three-agent talk among C, E, and T where C and E are the speaker and the hearer respectively while T is the translator.

Communication-unestablished: two agents speaking different single-languages cannot establish conversation, if it fails to find a translator among their common neighbors.

Success: in the hearer's memory, if there is a word the same as the one transmitted from the speaker or translated by the translator, a *success* happens. Therefore, a *success* could happen in both *pair-communication* (Fig. 2(a)) and *triangle-communication* (Fig. 2(b)). When a *success* happens, both the speaker and the hearer will clear their memories except the same word.

Failure: if the hearer does not have the same word transmitted from the speaker or translated by the translator in its memory, or the translator does not have the same word said by the speaker, a *failure* happens. Thus, a *failure* could happen in a *pair-communication* (Fig. 3(a)), or happen in a *triangle-communication* caused by the hearer (Fig. 3(b)), or happen in a *triangle-communication* caused by the translator (Fig. 3(c)). When a *failure* happens, the transmitted word will be added to the memory of the one who caused the failure.

Various forms of communications between two different language groups are summarized in Table 1. The gaming rule of the MLNG model is described by the following scheme, with two languages for illustration:

(1) Generate a population of N agents (nodes) connected in a certain topology, where the memory of every agent is empty.
(2) Randomly assign each agent with a label (C, E or T) with a certain probability, depending on the proportion of each language group. Set up their corresponding external vocabularies.
(3) At each iteration, randomly select a speaker and then select a hearer from the speaker's neighbors:

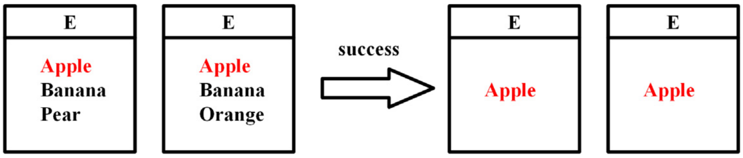
   a. If the speaker and the hearer belong to the same language group (they speak the same language), or if one of them is a T, then a *pair-communication* is established and this pair of agents can communicate with each other:

      i. Randomly pick a word from the speaker's memory if the memory is not empty; otherwise, randomly pick a word from its external vocabulary.

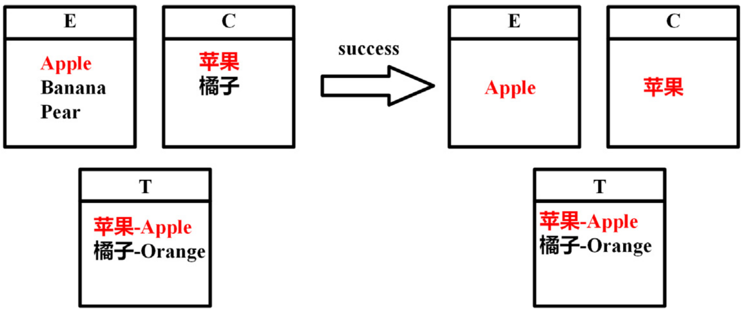
Table 1
Establishment of communications between different language groups.

Language group	C	E	T
C	√	① √ ② ×	√
E	① √ ② ×	√	√
T	√	√	√

Note: ① is the case that there is a translator T among the common neighbors of the speaker and the hearer, so a triangle-communication can be established. ② is the case that there is no translator T among the common neighbors of the speaker and the hearer, so no communication is possible. √ means that communication can be established while × means that communication cannot be established.



(a) A *success* in pair-communication.



(b) A *success* in triangle-communication.

Fig. 2. Situations when a *success* happens: (a) A *success* in *pair-communication*; (b) A *success* in *triangle-communication*.

ii. If the hearer has the same word in its memory, it is a *success* (Fig. 2(b)), so both the speaker and the hearer clear their memories except the common word; otherwise, it is a *failure* (Fig. 3(a)) and the hearer adds this word to its memory.

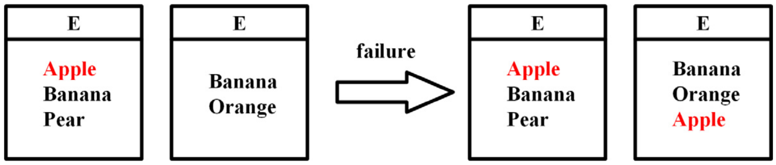
b. Otherwise, this pair of agents cannot communicate with each other directly. They start to search for a T from among their common neighbors. If there is no T in their common neighbors, the communication between this pair of agents cannot be established; but, if there is at least one T in their common neighbors, then randomly select a T as the translator, so a *triangle-communication* is established successfully:

i. Randomly pick a word from the speaker's memory if the memory is not empty; otherwise, pick a word from its external vocabulary.

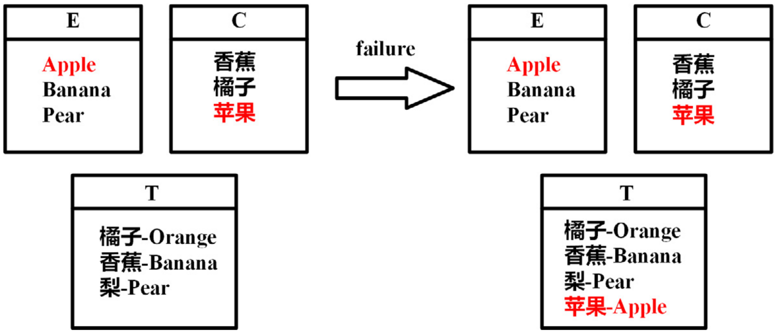
ii. If the translator T does not have this word in its memory, it is a *failure* (Fig. 3(b)), so the T adds this word to its memory. The hearer does nothing. But, if the T has this word in its memory while the hearer does not, it is a *failure* (Fig. 3(c)), so the hearer adds this word to its memory. If this word exists in both memories of the T and the hearer, it is a *success* (Fig. 2(b)), so both the speaker and the hearer clear their memories except keeping the common word.

(4) The process stops only if all the nodes have one and only one same word in their memories or the number of iterations reaches the pre-set threshold.

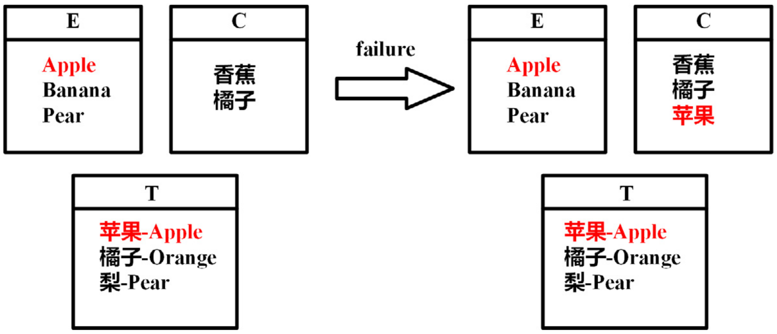
The job of the translator T in the *triangle-communication* is only to translate words for the two different language groups. Therefore, when a *success* happens to a *triangle-communication*, the translator T does nothing. However, in the situation of a *failure* caused by the translator T (as shown in Fig. 3(b)), the translator T will take responsibility of the failure and learns this word by adding this word to its memory.



(a) A *failure* in pair-communication.



(b) A *failure* in triangle-communication caused by *T*.



(c) A *failure* in triangle-communication caused by hearer.

Fig. 3. Situations when a *failure* happens: (a) A *failure* in *pair-communication*; (b) A *failure* in *triangle-communication* caused by *T*; (c) A *failure* in *triangle-communication* caused by the hearer.

3. Simulation results and analysis

3.1. Simulation settings

The MLNG has been simulated on five types of networks, so that major network natures, such as homogeneity, heterogeneity, small world and scale free, etc., can be covered. All these networks are studied on population sizes of 500 and 1000, respectively, to verify the consistency.

The performed simulations include random-graph networks (denoted as RG/P/N, where *P* is the connecting probability and *N* is the population size) generated by the Erdős–Rényi random-graph network algorithm [20] representing homogeneous networks. To study the small-world properties, the Watts–Strogatz (WS) small-world network algorithm [21] and the Newman–Watts (NW) small-world network algorithm [22] are employed. The corresponding two small-world networks are denoted by WS/K/RP/N and NW/K/RP/N, respectively, where $2K$ is the number of connected neighbors of the initial network, RP is the rewiring probability in the NW model or the link-adding probability in the WS model, and *N* is the population size. The study also includes scale-free networks (denoted as SF/m_0/m/N, where m_0 is the initial number of nodes, *m* is the number of edges being added to the network in each step, and *N* is the final population size) generated by the Barabási–Albert scale-free network algorithm [23] to investigate the heterogeneity and scale-free properties. To reveal the possible influence of triangles on the performances, the random triangle models [24] are simulated. The random triangle model is generated by starting from a fully-connected network with m_0 nodes, and then adding new nodes into the network step by step. When a new node is added, it connects to the network to form *m* new triangles. The way to form a triangle is to connect the new node to one existing node and one neighbor of the picked node, both at random. Here, a random triangle model is denoted as RTM/m_0/m/N, where *N* is the final population size.

Table 2
Network settings and results of simulations.

Notation	Network type	Number of nodes	Average degree	Average clustering coefficient	Average path length
RG/0.1/0.5K	Random-graph network with $P = 0.1$	500	49.90	0.1001	1.9061
RG/0.1/1K	Random-graph network with $P = 0.1$	1000	99.95	0.1001	1.9000
WS/25/0.2/0.5K	WS small-word network with $K = 25$ and $RP = 0.2$	500	50.00	0.4024	2.0074
WS/50/0.2/1K	WS small-word network with $K = 50$ and $RP = 0.2$	1000	100.00	0.4076	1.9149
NW/20/0.02/0.5K	NW small-word network with $K = 20$ and $RP = 0.02$	500	49.23	0.5054	2.0376
NW/40/0.02/1K	NW small-word network with $K = 40$ and $RP = 0.02$	1000	98.33	0.5113	1.9251
SF/25/25/0.5K	BA scale-free network with $m_0 = 25$ initial nodes and $m = 25$ new edges added at each step	500	48.64	0.1978	1.9268
SF/50/50/1K	BA scale-free network with $m_0 = 50$ initial nodes and $m = 50$ new edges added at each step	1000	97.39	0.1948	1.9044
RTM/50/13/0.5K	Random triangle model with $m_0 = 50$ initial nodes and $m = 13$ new triangles added at each step	500	49.14	0.1963	1.9433
RTM/50/27/1K	Random triangle model with $m_0 = 50$ initial nodes and $m = 27$ new triangles added at each step	1000	98.55	0.1588	1.9049

The network settings and topological results are summarized in Table 2. The reason for choosing those network parameters is to keep the average degree the same for a fair comparison.

In every network simulated, the agents are classified into three language groups including group C (Chinese speakers), group E (English speakers) and group T (Translators who can speak both Chinese and English). In order to make it simpler but obtain better observations on T, same proportions are assigned to C and E. The three groups, C, E, and T, are randomly distributed into every network.

In simulations, the size of the external vocabulary is 10,000 and the maximum number of iterations is 10,000,000, which is large enough for the 10 simulated networks to converge if the proportion of T (P_T) is not too small relatively. In order to decrease the randomness caused by the random distributions of the three language groups and the randomness of the naming game itself, every test was run for 20 trials independently and then they will be averaged. Namely, all the results are statistically averaged results from the 20 independent simulation trials.

3.2. Convergence process and analysis

Firstly, the effect of each agent (T, E, and C) is analyzed by observing the dynamics of the *number of total words (Nw)* and the *number of different words (Ndw)* in both global and local scopes during the process of convergence, where global scope means the whole network and local scope means inside each language group.

The proportion of T (P_T) is set to 0.1 and 0.5, representing the small P_T and the large P_T, respectively. It turns out that the convergence processes of all the 10 networks are quite similar. Thus, for clarity, only the random-graph network with 1,000 nodes is taken here as an example for analysis. Results of the other networks are shown in the file Supplementary Information (Figs. S1–S9.).

The simulation results are shown in Fig. 4. As can be seen, for both *Nw* and *Ndw*, every curve has the evolving period of first rising and then decreasing. One can regard the rising part of the curve as a period of exploring, during which a speaker is exploring the lexicon to find a proper word to name an object, which was done through the communications between agents with adding words from the external vocabulary to their memories. Similarly, the decreasing part of the curve can be regarded as a period of convergence, during which agents are changing towards consensuses by deleting inappropriate words from their memories.

Although the shapes of these curves shown in Fig. 4 are similar, there are some differences. It can be observed that the curves of the *Ndw* reach peaks ahead of those of the *Nw*. This is because the agents invent new words only if their memories are empty. When agents have at least one word in their memories, the *Ndw* would not increase anymore but failures would still happen, leading to the increase of the *Nw*. When inventing a word or *failure* happens, the new word will be added to both memories of the speaker and the hearer, the peak of *Ndw* should be half of the population size (if N denotes the number of agents, then the peak of *Ndw* should be around $0.5N$) [25]. Therefore, it is sure that the game finishes the exploring of new words first and the "true" convergence period starts after the number of total words reaches the peak.

The results are further analyzed in details from a local perspective. Although the group of T only occupies 20% of the population, the *Ndw* in T is larger than those in E and C most of the time, as shown in Fig. 4(b). The *Nw* in T is not smaller than those of the other two language groups in the early time of the process. The reason is that T can learn two languages and can communicate with all the agents in the population, implying that it has a better ability to explore and also to converge. Moreover, in groups E and C, the *Nw* increases extremely fast when T has large *Nw* and *Ndw* (the period between the point D and the point B in Fig. 4). This is because the translator T can help the communications between E and C. The more words the group T have, the stronger ability of translation they can provide. Another finding is that the group of T starts to converge earlier than the global one, while the groups E and C start to converge later than the global one. Although the proportion of the group of T is small (only 20%), they play a very important role in both exploring and convergence periods.

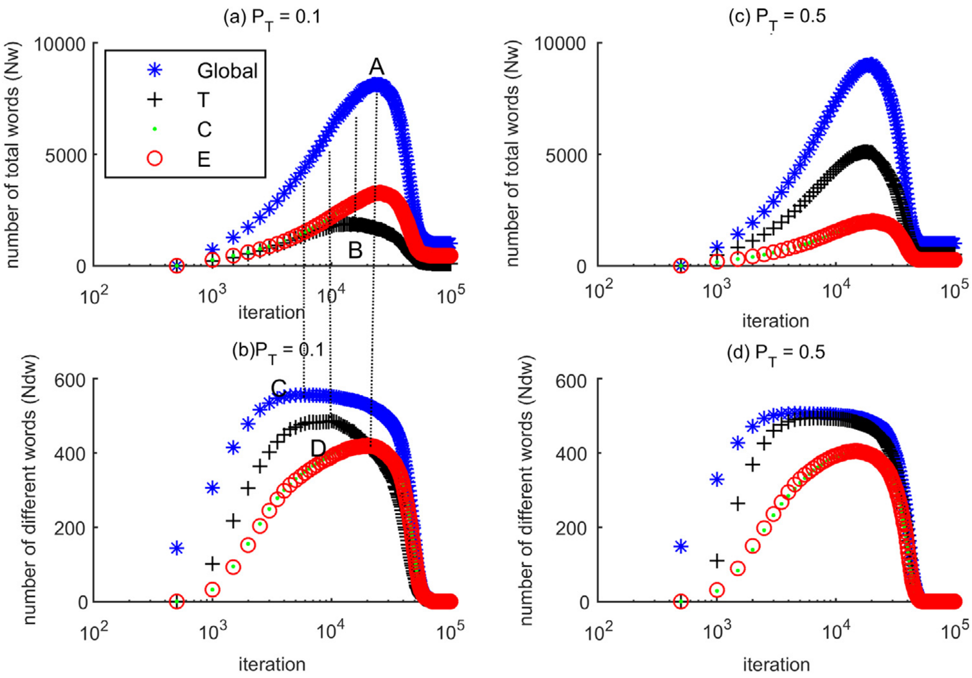


Fig. 4. The convergence processes in a random-graph network (RG/0.1/1K): (a) The *number of total words* (Nw), with $P_T = 0.1$; (b) The *number of different words* (Ndw), with $P_T = 0.1$; (c) The *number of total words* (Nw), with $P_T = 0.5$; (d) The *number of different words* (Ndw), with $P_T = 0.5$.

When the proportion of T increases to 50%, as shown in Fig. 4(c) and (d), the above conclusions remain although the convergence speed of each language group is getting closer to each other. Fig. 5 visualizes a trial performed on WS/50/0.2/1K with $P_T = 0.5$. In addition to the above conclusion, it can also be concluded that large-degree nodes start to explore (from 1,000th iteration to 20,000th iteration) which also converge (from 20,000th iteration to 50,000th iteration) faster than small-degree nodes. This is because there are more chances for nodes with larger degrees to participate in a game.

3.3. Communication ratios

In the MLNG model, there are two communication modes: *pair-communication* and *triangle-communication*. In this paper, *communication ratios* include the proportions of *pair-communication* (P_{pc}), *triangle-communication* (P_{tc}) and *communication-unestablished* (P_{cu}) through the whole process of the game.

The P_T is the only parameter that can influence the proportion (or probability in a single iteration) of *pair-communication*. Since P_{pc} denotes the probability of choosing two agents who can communicate with each other directly, it can be calculated as follows:

$$P_{pc} = 1 - (1 - P_T)^2 + 2 \times \left(\frac{1 - P_T}{2}\right)^2 = 1 - \frac{(1 - P_T)^2}{2}. \quad (1)$$

As to P_{tc}, it is related to three factors: P_T, the average degree ad and the average clustering coefficient cc. Knowing these three factors, the following procedures are to estimate P_{tc}.

Step 1: Calculate the probability of not running *pair-communication*, $\frac{(1-P_T)^2}{2}$.

Step 2: Pick a speaker and take all its neighbors as a sub-network (excluding the speaker). If this sub-network is regarded as a random graph, one can calculate an approximate connecting probability p between a pair of neighbors. Among the neighbors, the number of edges is about $\binom{ad}{2} \cdot cc$. Therefore, cc can be regarded as an approximation of p.

Step 3: Pick a hearer from the sub-network in Step 2 and then calculate P_{tc}. To find a T as the third node of the triangle, since all the agents in this sub-network are the neighbors of the speaker, one only needs to consider the neighbors of the hearer. For the hearer, there are approximately $cc \cdot (ad - 1)$ neighbors and the probability of having at least one T is

$$P_{ET} \approx 1 - (1 - P_T)^{cc\cdot(ad-1)}. \quad (2)$$

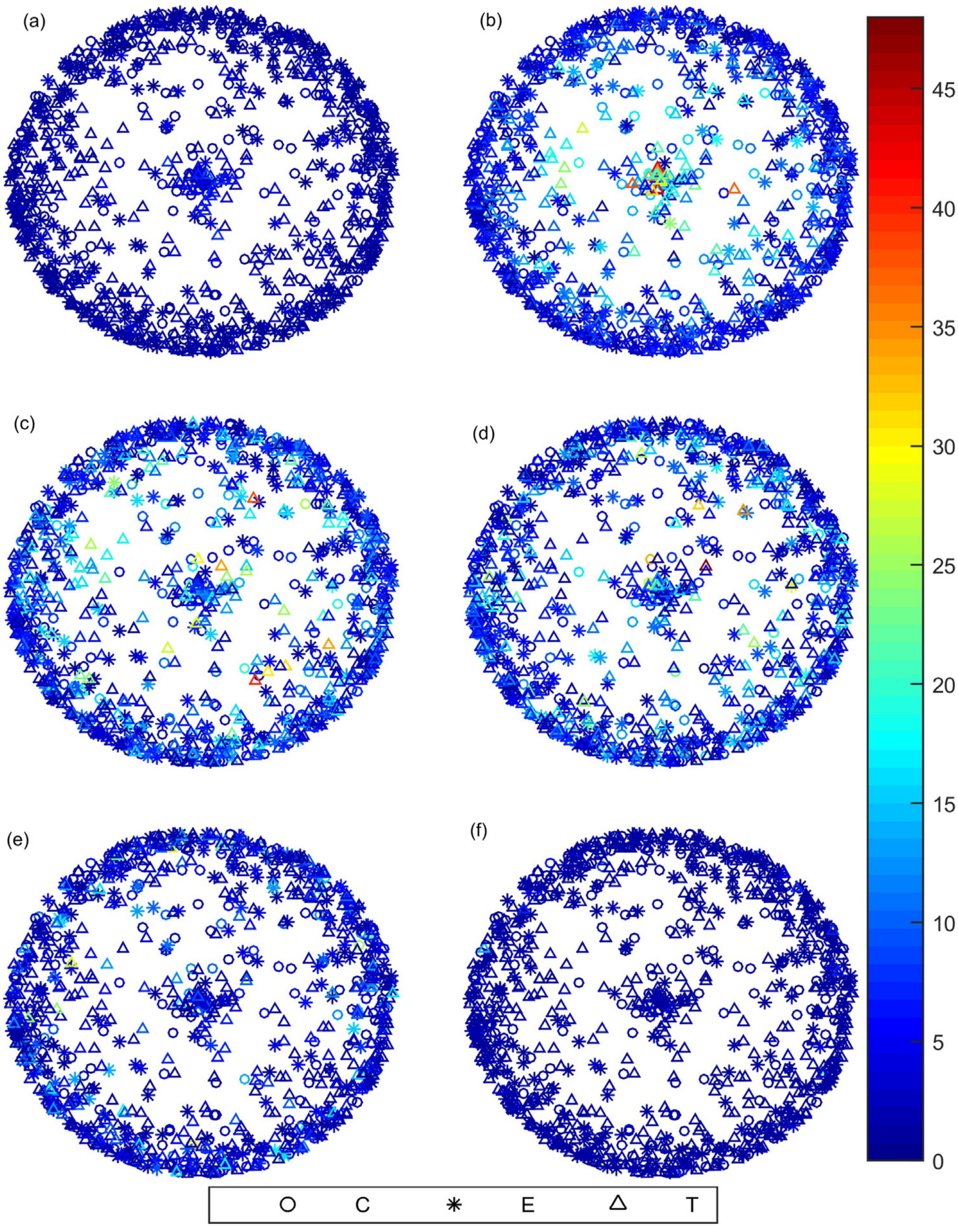


Fig. 5. Visualization of the MLNG performed on a scale-free network (SF/50/50/1K, $P_T = 0.5$). Nodes are distributed within a circle, so that those near the center have larger degrees. The color indicates the number of words in each node. (a) Iteration = 1000, mean = 1.41; (b) Iteration = 10,000, mean = 6.473; (c) Iteration = 20,000, mean = 7.05; (d) Iteration = 30,000, mean = 6.457; (e) Iteration = 40,000, mean = 2.925; (f). Iteration = 50,000, mean = 1.116. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

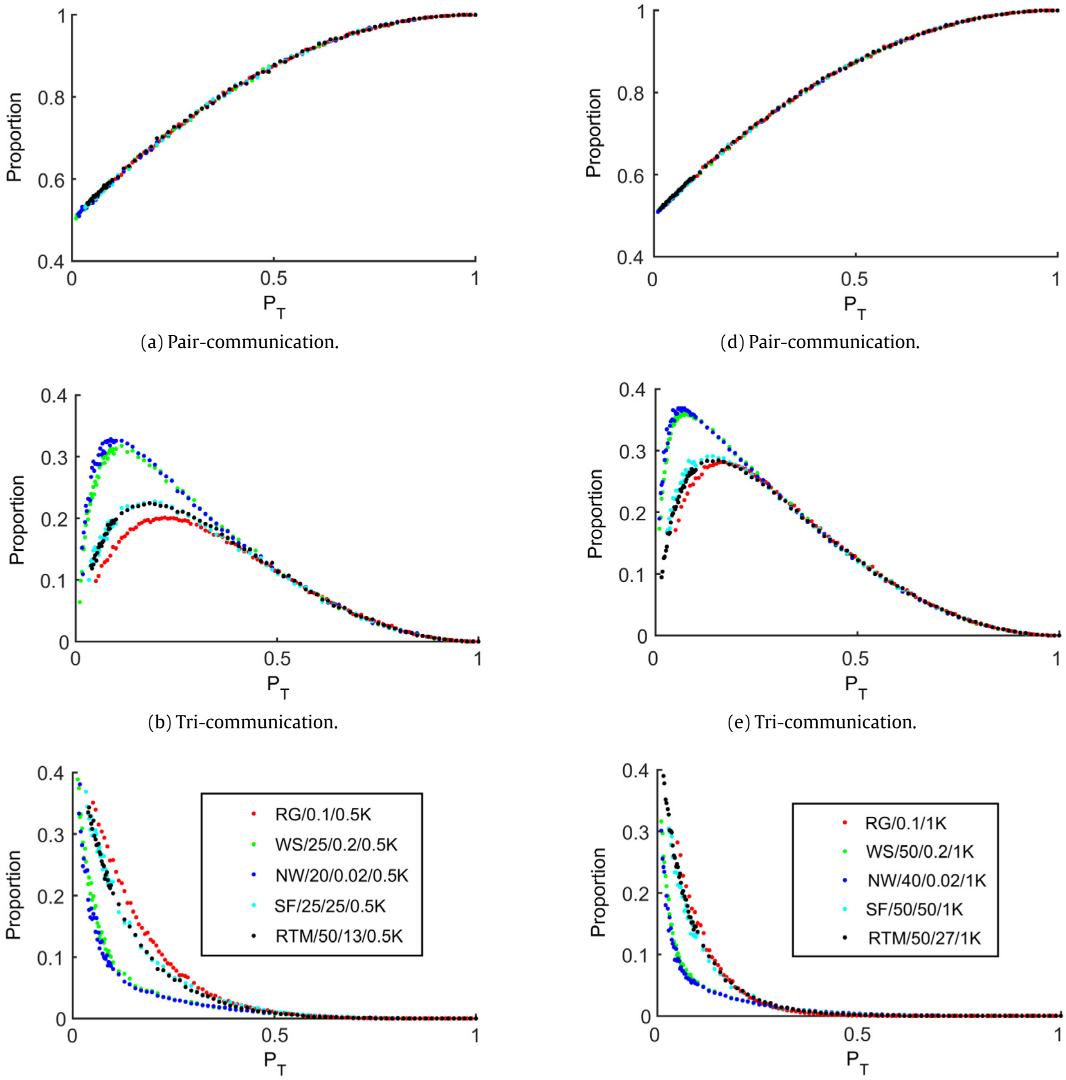
Here, $cc \cdot (ad - 1)$ is defined as the *triangle-communication coefficient*, C_t, which depends on the network itself. Then, P_{tc} is approximately calculated as

$$P_{tc} = (1 - P_{pc}) \cdot P_{ET} \approx \frac{(1 - P_T)^2}{2} \cdot \left[1 - (1 - P_T)^{C_t}\right]. \tag{3}$$

After obtaining P_{pc} and P_{tc}, P_{cu} can be easily obtained by

$$P_{cu} = 1 - P_{pc} - P_{tc} \approx \frac{(1 - P_T)^{2+C_t}}{2}. \tag{4}$$

Fig. 6 shows the results of the *communication ratios* tested on the 10 networks. Fig. 7 shows the comparisons between the results of calculated (predicted) *communication ratios* and the measured *communication ratios*. From Fig. 7, one can see



(a) Pair-communication. (d) Pair-communication.

(b) Tri-communication. (e) Tri-communication.

(c) Communication-unestablished. (f) Communication-unestablished.

Fig. 6. The results of *communication ratio* in 10 networks: (a) and (d) show the results of *pair-communication*; (b) and (e) show the results of *triangle-communication*; (c) and (f) show the results of *communication-unestablished*.

that the calculated (predicted) curves fit the measured curves very well for all the 10 networks, confirming that formulas (1), (3) and (4) can be used as estimations of P_{pc}, P_{tc} and P_{cu}. This is valid not only for random-graph networks but also for the small-world networks, scale-free networks and the random triangle models. From (4), one can see that the average degree and the average clustering coefficient are the two main features that influence the *communication ratios* most significantly.

In order to have a better understanding of the behaviors of the agents, a further test on the *triangle-communication* is performed. When P_T is very small, if a *pair-communication* cannot be established, it will more likely go to the *communication-unestablished* rather than to the *triangle-communication*, because it is very difficult to find a T to do the translation. This is the period for which the proportion of *communication-unestablished* is high. Then, as P_T increases, this situation is improved significantly and more *triangle-communications* are established. However, as P_T becomes larger and larger, more *pair-communications* are established and, consequently, the proportion of *triangle-communication* becomes smaller and smaller.

3.4. Convergence speed study

A test studying on the relationship between the convergence speed and the proportion of T (P_T) has also been performed. The number of iterations is examined when the game reaches the global consensus, which reflects the convergence speed of the game.

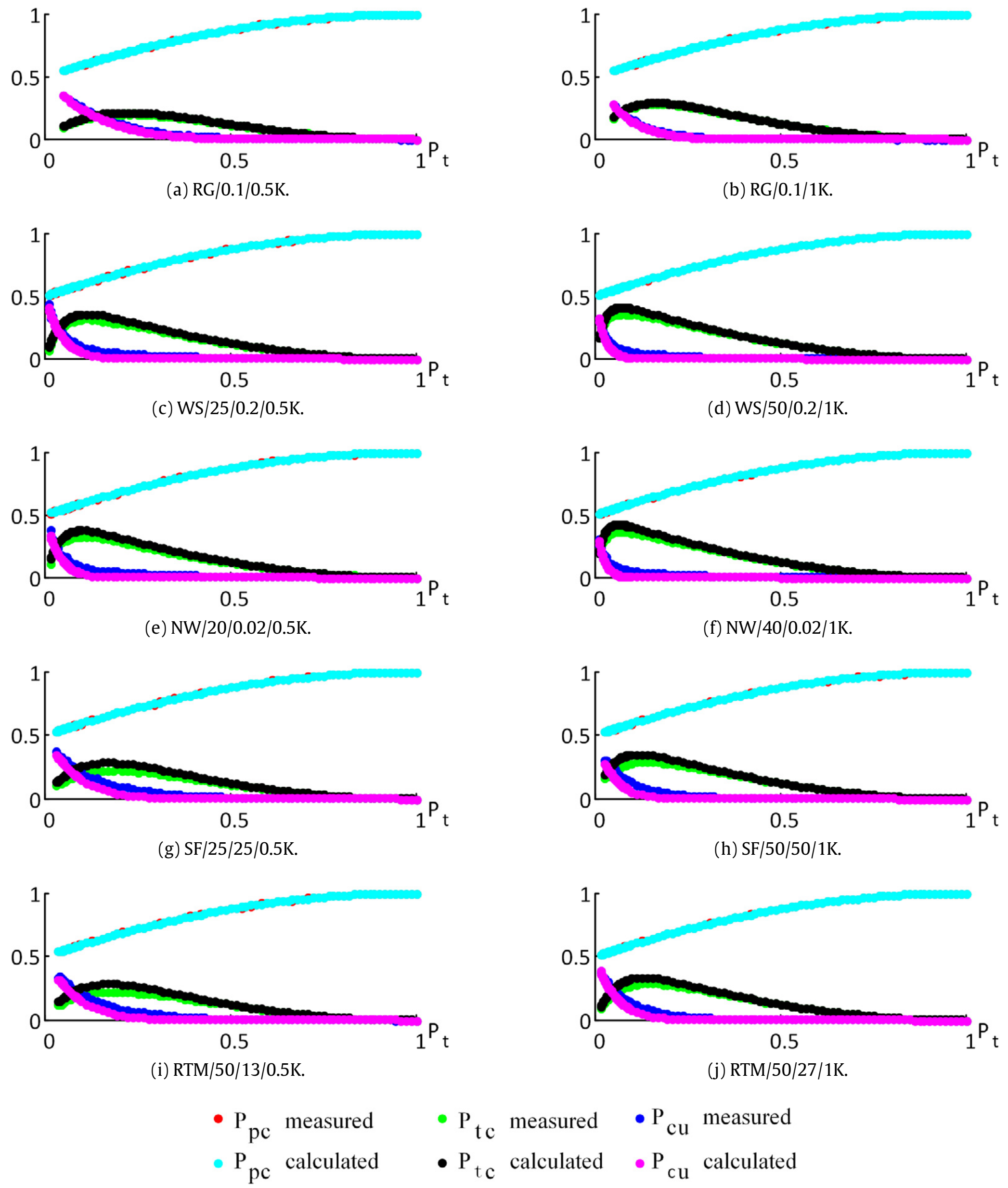


Fig. 7. Results of calculated *communication ratios* and the measured *communication ratios* in 10 networks.

When running the test, it was found that the convergence speed varies a lot with a lower proportion of T, while it does not change much when the proportion of T is high. Thus, the part of lower proportion of T is divided into some small intervals. However, when P_T is too small, the game may not converge within the maximum number of iterations. In such cases, the average convergence time makes no sense. Therefore, the average convergence time is kept to start from the smallest P_T, where there is no unsuccessful convergence happening through the total of 20 trials, which were repeated for all larger P_T cases.

The pre-set values of P_T for the simulation and the resulting values of P_T after the simulation are both shown in Table 3, where only a small part of P_T is listed because the part larger than 0.3 will never be affected.

The simulation results are shown in Fig. 8. As can be observed, the main features of the relationship between the convergence time and the proportion of T (P_T) are the same among all the 10 different networks: the convergence time

Table 3
The settings for convergence speed study.

Notation	Pre-set proportion of T (P_T)	Resulting proportion of T (P_T)
RG/0.1/0.5K	Setting 1	Unchanged
RG/0.1/1K	Setting 1	Unchanged
WS/25/0.2/0.5K	Setting 2	Unchanged
WS/50/0.2/1K	Setting 2	Unchanged
NW/20/0.02/0.5K	Setting 2	$0.0151 \le P_T < 0.3$: 33 intervals
NW/40/0.02/1K	Setting 2	Unchanged
SF/25/25/0.5K	Setting 2	$0.0331 \le P_T < 0.3$: 26 intervals
SF/50/50/1K	Setting 2	$0.0280 \le P_T < 0.3$: 28 intervals
RTM/50/13/0.5K	Setting 2	$0.0383 \le P_T < 0.3$: 24 intervals
RTM/50/27/1K	Setting 2	$0.0151 \le P_T < 0.3$: 33 intervals

Setting 1: $0.05 \le P_T < 0.3$: 35 intervals and $0.3 \le P_T \le 1.0$: 75 intervals; *Setting 2:* $0.01 \le P_T < 0.3$: 35 intervals and $0.3 \le P_T \le 1.0$: 75 intervals.

decreases with increasing P_T, which means that the more translators exist, the faster the game will reach the state of global consensus.

More precisely, with t_c denoting the convergence time, the relationship between $\log(t_c)$ and $\log(P_T)$ are approximately linear. Therefore, the relationship between the convergence speed and P_T follows a power-law-like relation. This finding suggests a possible mathematical model for approximating the relationship between the convergence speed and P_T in an MLNG model, which will help a lot in predicting the convergence speed when knowing the proportion of T. Roughly,

$$t_c \approx a \cdot P_T^{-\gamma} \tag{5}$$

where a is the coefficient determined by the convergence time obtained when there are only single-language speakers in the conventional naming game model.

In general, comparing the same type of networks, 500-node networks can converge faster than 1000-node networks. But, different types of networks perform differently. As can be seen from Fig. 8, random-graph networks, scale-free networks, and random triangle models perform similarly. The curves of these three types of networks have similar small gradients, implying that they hold a small degree exponent γ in (5). When it comes to the small-world networks, the gradients of their curves are relatively large (with a large γ in (5)). Observing Table 2 and Fig. 8, from the feature point of view, one can conclude that the networks with larger average clustering coefficients (cc) have larger γ in (5).

To further investigate the correlation between γ and cc, a set of networks with different cc are generated, for which the degree of each agent remains unaltered. These networks, referred to as dk-graphs, are generated by performing the dk-targeting rewiring algorithm with $d = 2.1$ [24] on all the 500-node networks shown in Table 3. The generating procedure of dk-graphs can be found in the Supplementary Information. Formula (5) is used to fit the relation between t_c and P_T for each plot, as shown in Figs. S18–S26 in the Supplementary Information. Fig. 9 shows the relationship between γ in (5) and cc of the simulated dk-graphs. One can observe that, when cc is small, γ almost keeps the same with little variation. However, γ increases significantly when cc is large. To a certain degree, the exponent γ reflects how significantly the increase of P_T would speed up the convergence process. When cc is small, there are limited complete triangles for the T to establish *triangle-communications*. Consequently, it limits the ability of T in speeding up the convergence, resulting in a small γ. On the contrary, when cc is large, T will have a great impact on the speed of convergence because there are more *triangle-communications* for T to play an effective role.

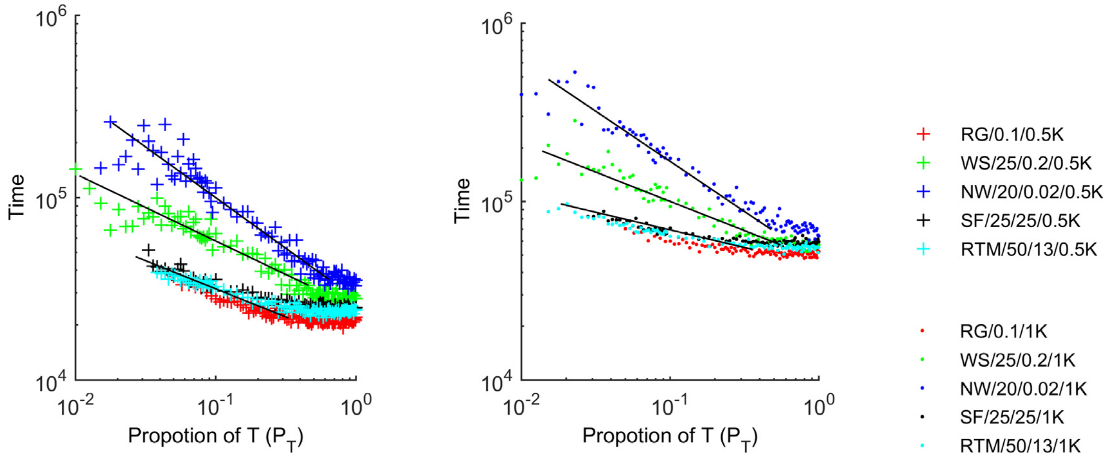
3.5. Maximum number of different words (Mdw) and maximum number of words (Mw)

The *Maximum number of different words* (*Mdw*) specifies the requirement of memory size and the *maximum number of words* (*Mw*) shows how many memory resources are actually used throughout the process of the game. The dynamics of the *maximum number of different words* and the *maximum number of words* are now examined. It is noticed that the features shown on different networks are similar. For clarity, only WS small-world networks are discussed, as an example for analysis. The results of the other networks can be shown in Figs. S20–S17 of the Supplementary Information.

Fig. 10 shows *the maximum number of different words* in the whole network for different P_T and Fig. 11 shows the *maximum number of different words* in different language groups. As can be seen from Fig. 10, generally the curves would not simply increase or decrease all the time with the increase of P_T, but they perform similarly with the P_{tc} shown in Fig. 6.

As discussed in Section 3.2, the *Mdw* should be around 0.5 N. However, when a *failure* caused by a translator happens in a *triangle-communication* (as shown in Fig. 10(b)) and if there is no word yet in the memory of the speaker, the word invented by the speaker will be added to the memory of the translator.

The repeating of same hearers or translators in *failure*, when inventing a word, will increase the *Mdw*. When P_T is small, there is a relatively high chance for the same T to be chosen as the translator in *triangle-communication*, and with a larger P_{tc}, the more chance the same T will be chosen. Therefore, the relationship between the *Mdw* and P_T is similar with that between P_{tc} and P_T which can also be observed by comparing Figs. 7 and 10.



(a) 500 nodes. (b) 1000 nodes.

Fig. 8. Relationships between the convergence time and the proportion of T (P_T) (log–log): (a) Results on 500-node networks; (b) Results on 1000-node networks.

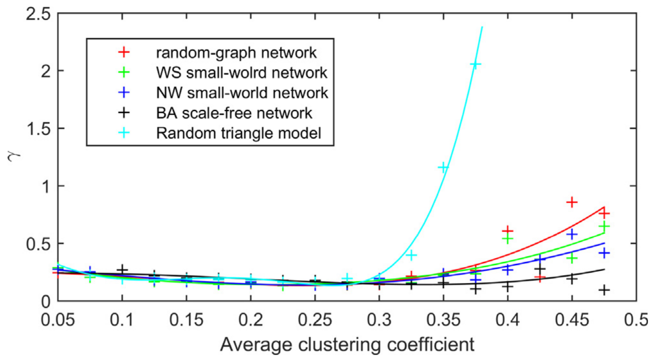


Fig. 9. Relationships between γ and the average clustering coefficient. The legends indicate the topologies of the original networks used to generate the *dk*-graphs.

Observing Fig. 11, one can see that the *Mdw* in various language groups are different but they are all smaller than the global one. If the proportions of T, E and C are the same (when P_T is about 0.33), the *Mdw* of T is larger than those in E and C, meaning that the group of T's requires larger memory sizes.

Memory allocation is an important issue in naming game models and has been studied in e.g. [3]. From a global point of view, if one considers all the agents to be the same and assigns them with the same memory size according to Fig. 10, then the proportion of language group T, P_T, would significantly influence the requirement of the memory size. One may thereby regard this assignment rule as a *global memory allocation rule*.

On the other hand, one can assign different memory sizes to different language groups according to Fig. 11, because their requirements of memory sizes are different from each other. One, therefore, can regard this assignment rule as a *local memory allocation rule*. Obviously, the local allocation rule is a more reasonable choice for saving the memory resources.

The relationship between the *maximum number of words* (*Mw*) in different language groups and the proportion of T (P_T) is shown in Fig. 12. The basic rule is that the *Mw* in T increases as P_T increases, which is opposite to those in E and C. An interesting finding is that, on the left-hand side of the crossover points of the curve T and the curve E (or C) (the left-hand side of the dash line in Fig. 12), the curve T and curve E (or C) change nonlinearly. However, the curves become more linearly on the right-hand side. It is because, when P_T is small, the increase of P_T would not only influence the number of agents in each language group but also on the *communication ratios* significantly. However, when P_T is relatively large, the change of *communication ratios* can be negligible, which leads to the more linear curves.

A similar phenomenon can be found on the curve of the global *Mw*. The difference is that the global *Mw* maintains a high level and does not change much after the nonlinear part, which makes sense because:

(1) The peaks of the *number of words* of language groups, as shown in Fig. 4, become more synchronizing when P_T becomes larger.

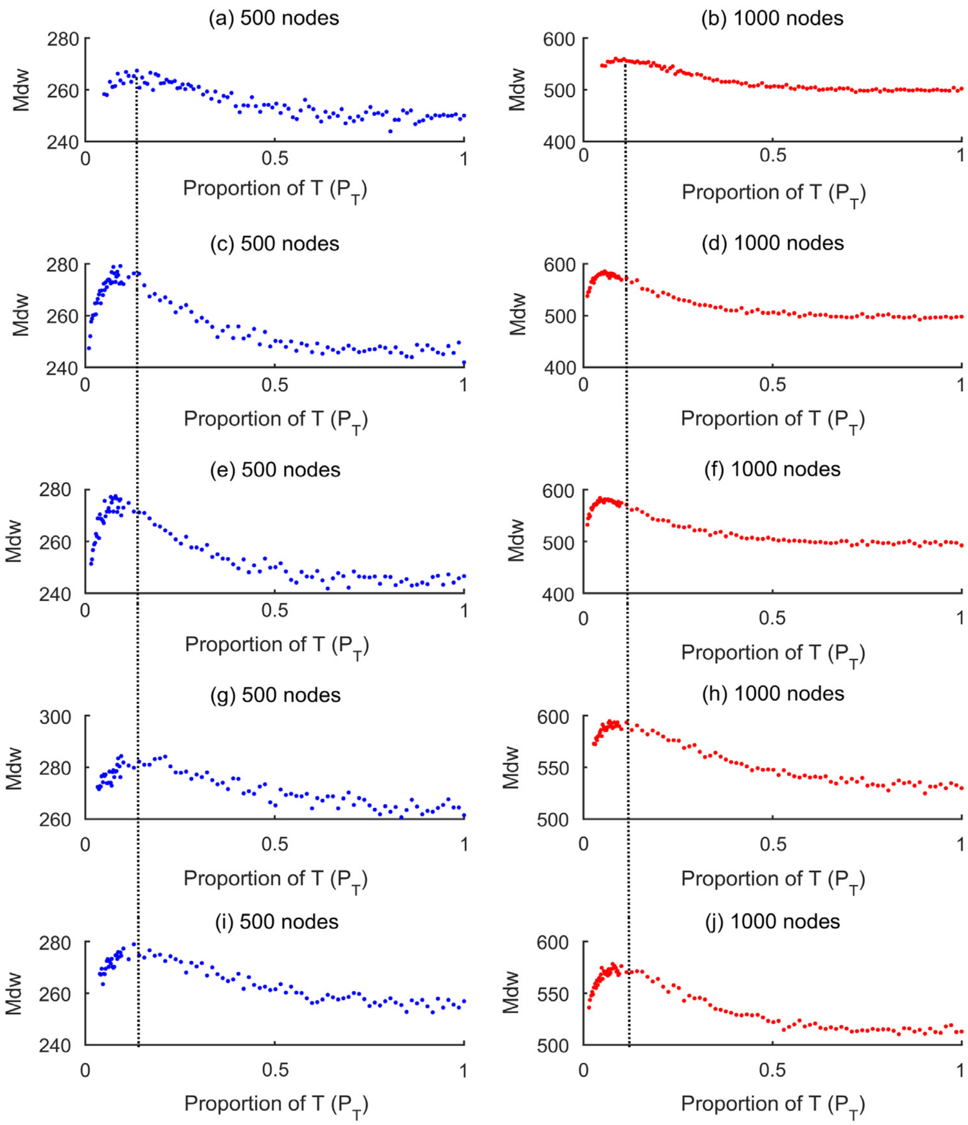


Fig. 10. Relationships between the *maximum number of different words* (Mdw) and the proportion of T (P_T): (a) RG/0.1/0.5K; (b) RG/0.1/1K; (c) WS/25/0.2/0.5K; (d) WS/50/0.2/1K; (e) NW/20/0.02/0.5K; (f) NW/40/0.02/1K;(g) SF/25/25/0.5K; (h) SF/50/50/1K; (i) RTM/50/13/0.5K; (j) RTM/50/27/1K.

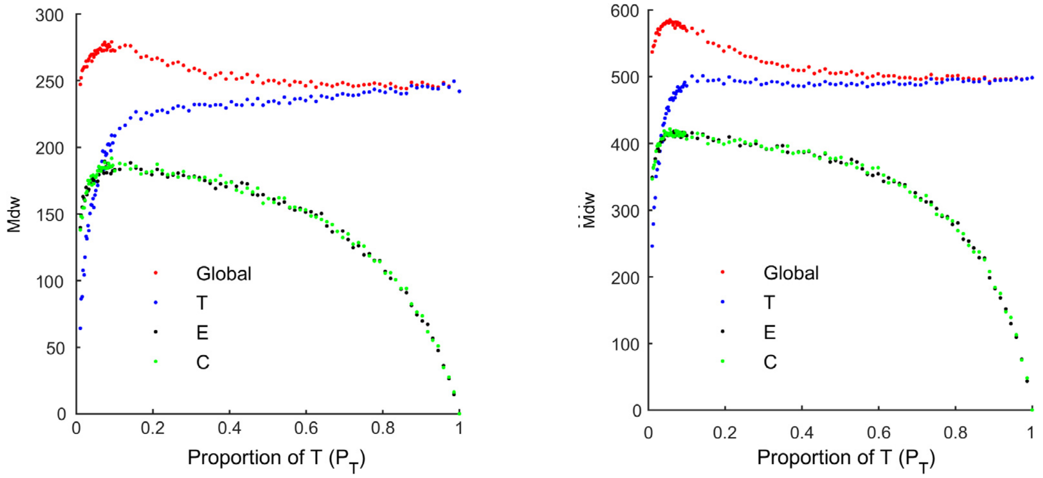
(2) The curve of the *maximum number of words* in each language group shown in Fig. 12 changes more linear when P_T is large.

These two reasons make the global *Mw* equals to the linear combination of the *Mw* in different language groups. It can also be concluded that, with the same requirement of the memory size, the language group T demands more memory resources than groups E and C.

4. Conclusions

In this paper, a realistic model called the multi-language naming game (MLNG) is developed, in which there are multiple languages unlike one single language in the traditional naming games. The new model has been studied through extensive computer simulations on several types of typical population networks.

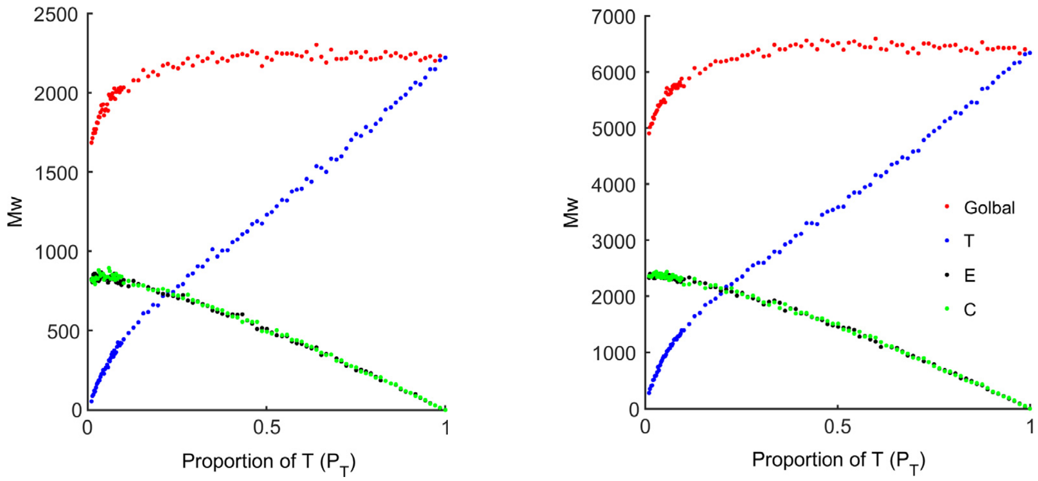
The evolutionary process of the naming game has been analyzed, leading to an effective method for estimating the probabilities of different communication modes using the network topological features such as the average degree, average clustering coefficient, and proportion of translators in the language groups. It has been shown that the MLNG has an interesting power-law-like relation between the convergence speed and the proportion of translators in the population.



(a) 500 nodes.

(b) 1000 nodes.

Fig. 11. Relationships between the *maximum number of different words* (Mdw) in each language group and the proportion of T (P_T) in WS small-world networks: (a) WS/25/0.2/0.5K; (b) WS/50/0.2/1K.



(a) 500 nodes.

(b) 1000 nodes.

Fig. 12. Relationships between the *maximum number of words* (Mw) in each language group and the proportion of T (P_T) in WS small-world networks: (a) WS/25/0.2/0.5K; (b) WS/50/0.2/1K.

Moreover, the memory size requirement of each agent has been examined, showing that different agents require different memory sizes. It thus suggests a *local memory allocation rule* which can better allocate different memory sizes to different agents for saving memory resources.

The new findings should provide new insights to further studies of language emergence and evolution, as well as computer languages for artificial intelligence (AI) research. There are two possible directions for future work: one is to study, based on the MLNG, the evolution of multiple languages that includes both birth and death processes; the other is to apply the MLNG to investigate self-organized systems toward AI language design and utilization.

Acknowledgment

The work described in this paper was partially sponsored by Huawei Innovation Research Program.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.physa.2017.12.124.

References

[1] L. Steels, A self-organizing spatial vocabulary, Artif. Life 2 (3) (1995) 319–332.
[2] A. Baronchelli, A gentle introduction to the minimal naming game, Belg. J. Linguist. 30 (1) (2016) 171–192.
[3] A. Baronchelli, et al., Sharp transition towards shared vocabularies in multi-agent systems, J. Stat. Mech. Theory Exp. 2006 (06) (2006) P06014.
[4] W. Wang, et al., Agreement dynamics of finite-memory language games on networks, Eur. Phys. J. B 60 (4) (2007) 529–536.
[5] A. Baronchelli, Role of feedback and broadcasting in the naming game, Phys. Rev. E 83 (4) (2011) 046103.
[6] B. Li, G. Chen, T.W. Chow, Naming game with multiple hearers, Commun. Nonlinear Sci. Numer. Simul. 18 (5) (2013) 1214–1228.
[7] S.K. Maity, et al., Emergence of fast agreement in an overhearing population: The case of the naming game, Europhys. Lett. 101 (6) (2013) 68004.
[8] Y. Lou, G. Chen, Analysis of the "naming game" with learning errors in communications, Sci. Rep. 5 (2015) 12191.
[9] G. Verma, A. Swami, K. Chan, The impact of competing zealots on opinion dynamics, Physica A 395 (2014) 310–331.
[10] Y. Lou, et al., Local communities obstruct global consensus: Naming game on multi-local-world networks, Physica A 492 (2018) 1741–1752.
[11] Y. Lou, G. Chen, J. Hu, Communicating with sentences: A multi-word naming game model, Physica A 490 (2018) 857–868.
[12] D. Lipowska, A. Lipowski, Naming game on adaptive weighted networks, Artif. Life 18 (3) (2012) 311–323.
[13] G. Fu, Y. Cai, W. Zhang, Analysis of naming game over networks in the presence of memory loss, Physica A 479 (2017) 350–361.
[14] G. Fu, W. Zhang, Naming game with biased assimilation over adaptive networks, Physica A (2017).
[15] A.M. Thompson, B.K. Szymanski, C.C. Lim, Propensity and stickiness in the naming game: Tipping fractions of minorities, Phys. Rev. E 90 (4) (2014) 042809.
[16] F. Palombi, S. Toti, Topological aspects of the multi-language phases of the naming game on community-based networks, Games 8 (1) (2017) 12.
[17] D. Lipowska, Naming game and computational modelling of language evolution, Comput. Methods Sci. Technol. 17 (1–2) (2011) 41–51.
[18] A. Lipowski, D. Lipowska, Computational approach to the emergence and evolution of language-evolutionary naming game model, e-print, 2008.
[19] L. Pucci, P. Gravino, V.D. Servedio, Modeling the emergence of a new language: Naming game with hybridization, in: International Workshop on Self-Organizing Systems, Springer, 2013.
[20] P. Erdos, A. Wi, On random graphs I, Publ. Math. Debrecen 6 (1959) 290–297.
[21] D.J. Watts, S.H. Strogatz, Collective dynamics of 'small-world' networks, Nature 393 (6684) (1998) 440.
[22] M.E. Newman, D.J. Watts, Renormalization group analysis of the small-world network model, Phys. Lett. A 263 (4) (1999) 341–346.
[23] A.-L. Barabási, R. Albert, Emergence of scaling in random networks, Science 286 (5439) (1999) 509–512.
[24] C. Orsini, et al., Quantifying randomness in real networks, Nat. Commun. 6 (2015) 8627.
[25] B. De Vylder, K. Tuyls, How to reach linguistic consensus: A proof of convergence for the naming game, J. Theoret. Biol. 242 (4) (2006) 818–831.