A Multi-word-agent Autonomous Learning Model for Regulating Word Combination Strength: Appendix

Jinfeng Yang[1], Yi Guan[*1], Xishuang Dong[2]
[1]*School of Computer Science and Techonlogy, Harbin Institute of Technology, Harbin, 150001, P.R. China*
[2] *School of Computer and Information Technology, Xinyang Normal University, Xinyang, 464000, P.R. China*
fondofbeyond@163.com, guanyi@hit.edu.cn, dongxishuang@gmail.com

1. Inspirations from immune systems

Immune systems build two levels of barriers: the innate (non-specific) immune system and the adaptive (specific) immune system. The proposed AIS model is inspired by the adaptive immune system due to its ability to recognize specific Ags. The adaptive immune system is a composite of numerous lymphocytes and the immune environment in which lymphocytes interact with each other. In this model, the immune environment is simulated as an $M \times M$ grid. The lymphocytes reside in each site of the grid and can move freely to the adjacent sites.

B cells are important lymphocytes in the adaptive immune system. Different B cells may have different concentrations. Higher concentrations of B cells indicate that these B cells are more important because they need to recognize more Ags. B cells can recognize specific Ags with their receptors and can then be stimulated by the Ags. The receptors of B cells have a Y-shaped structure[1], with two variable regions at the tips of the Y. In the variable region, there exist two specific, unique topography sites, namely paratopes and idiotopes. The paratopes are responsible for recognizing Ags, and the idiotopes can function as antigens. Thus, the paratopes of one B cell can also recognize the idiotopes of another B cell. The two variable regions at the tips of the Y are identical. For simplicity, in the proposed model, the left tips of the Y are idiotopes and the right tips are paratopes. Figure 1 shows the simplified design of B cell receptors. In the model, B cells represent words and are modeled as BWAs, and the B cell concentration represents the word's frequency. The B cell receptors represent word properties (or features). This model aims to regulate the dependency relation strength between words; therefore, the dependency features extracted from annotated head-dependent pairs are used as word properties and are grouped into dependent properties, corresponding to idiotopes, and head properties, corresponding to paratopes.

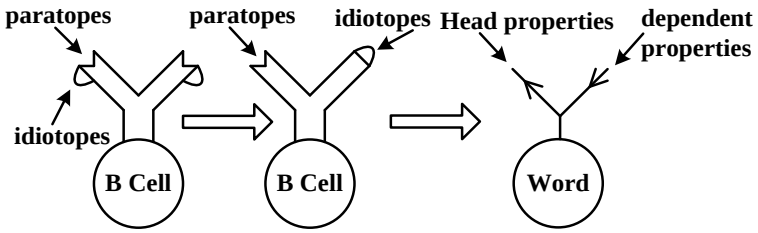

Figure 1. The design of B cell receptors and word properties

In the immune response, the only role of Ags is to match and stimulate B cells and get killed. Ags match the B cells' paratopes with their unique set of antigenic determinants, also called epitopes. In the proposed model, Ags also function as words, and epitopes are the words' head properties.

The interactions between B cells and Ags or other B cells are determined by the affinity between paratopes and epitopes or idiotopes. In the proposed model, the combination strength is calculated based on the similarity between paratopes and epitopes or idiotopes, accumulating weights of the matched properties. The initial weights of word properties can be set to zero or to random values.

In idiotypic immune network theory, the idiotopes of one B cell can match the paratopes of another B cell. This type of interaction results in the network of B cells. The immune network is not structured randomly, but it is topologically a small world network [2]. According to the simulation, a power-law degree-distribution can emerge[3]. Similar to an immune network in terms of complexity, the language network is also a complex network[4]. Due to the similarity between the idiotypic immune network and the language network, the language network built from a dependency Treebank simulates the idiotypic immune network in the proposed model.

In an idiotypic immune network, the idiotypic interaction should not spread to the entire network, and it may stop based on certain criterion, such as the maximum idiotypic level[5]. Spreading activation is used in cognitive psychology to model the fan-out effect[6] and is also used as the memory mechanism of humans using a language network[4]. In this AIS model, spreading activation is employed as the idiotypic interaction mechanism as follows. As shown in Figure 2, the language network is simulated as an immune network in which nodes are BWA and weighted arcs are dependency relations with various strength. When an AgWA stimulates a BWA (e.g., BWA 1), the BWA is activated and is assigned an initial active value (e.g., 3). This initial active value is called the activation level, which determines the spreading depth and can spread along the inverse direction of arcs. The activated BWA can stimulate another dependent BWA (e.g., BWA 3) that has the greatest affinity in the local site and then transfer its weakened active value (e.g., 2) to the second activated BWA. This process of spreading activation continues until the active value is weakened to zero.

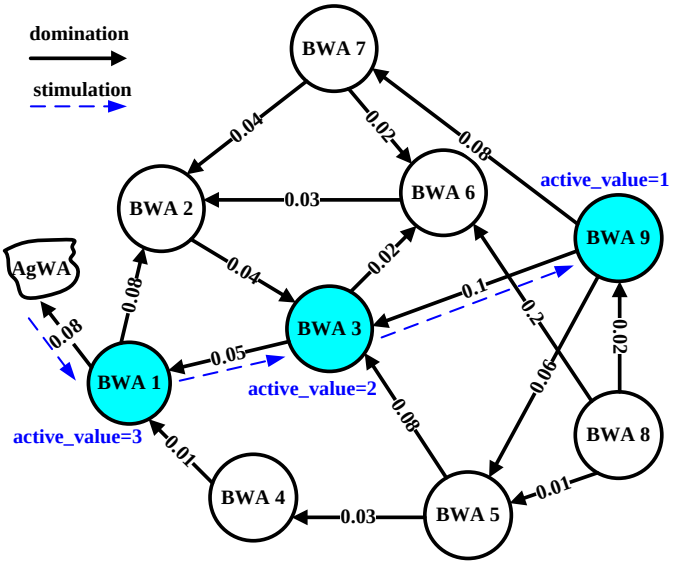

Figure 2. The process of spreading activation in the artificial immune network of this model

During the immune response period, the concentrations of B cells may change because the stimulated B cells can self-reproduce, a phenomenon called clonal expansion. In this model, clonal expansion is applied to generate a number of candidates. The number of clonal candidates is a parameter in our model. Hypermutation is the most important mechanism experienced by the offspring, and it leads to the generation of more powerful B cells. In our model, hypermutation is introduced for the matched properties between the paratopes of the offspring and the epitopes of the recognized Ag. The result of hypermutation is a group of random increments of the weights of the matched properties. The mutation is inversely proportional to the weight of the property or the affinity with the matched Ag (the higher the affinity or the weight of the property is, the lower the mutation rate is[7]).

After hypermutation, some offspring with increased affinity are reserved and undergo differentiation to generate plasma cells or memory cells. This process is called affinity maturation. Affinity maturation is simulated as the process wherein the best offspring are selected. A fitness function is designed for the mutated offspring, and the best offspring can be determined. The best offspring is reserved, and the others are eliminated. If the best offspring is better than the parent, then the parent is replaced by the reserved best offspring.

Based on these inspirations, Agents, including BWAs and AgWAs, and the MWAALM environment are designed. The components of adaptive immune theory and their counterparts in MWAALM are summarized below in table 1.

Table 1. The components of the immune system and their counterparts in MWAALM

adaptive immune theory	MWAALM
Immune environment	An $M \times M$ grid
B cells	B cell word agents
Antigens	Antigen word agents
Idiotopes of B cell receptors	Dependent properties of words
Paratopes of B cell receptors	Head properties of words and their weights
Epitopes of antigens	Dependent properties of words
Concentration of B cells	Word frequency
Affinity	Combination strength
Immune network	Language network
Idiotypic interaction	Spreading activation
Clonal expansion	Reproduction of offspring as candidates
Mutation	Generation of increments of the matched properties' weights
Affinity maturation	Selection of the best mutated offspring

2. Multi-word-agent autonomous learning model

2.1. Outline

The proposed model, MWAALM, is inspired by the intrinsic consistencies between words in language systems and B cells in immune systems, and it aims to regulate the combination strength between words. In this model, words are simulated as B cells or antigens. Then, adaptive immune theories, the clonal selection principle and the immune network are introduced to design this AIS-based learning model. As an agent-based modeling method, CA is a natural application for modeling cellular systems. Being a generic framework, AOC offers a new computing paradigm that makes use of autonomous entities in solving computational problems and in modeling complex systems. In this research, the proposed model is constructed using CA in the AOC framework. The model consists of three components: a group of autonomous word agents (BWAs and AgWAs), an artificial immune environment in which word agents reside and interact with each other, and a system objective function.

To train and evaluate this AIS-based model, a Chinese dependency Treebank is divided into a training set and a test set. The learning process of the model is equivalent to the artificial immune response. The whole process of this AIS model includes three stages. In the initialization stage, the immune environment and B cell word agents are initialized. The immune environment is initialized as an $M \times M$ grid. B cell word agents are built from the training set, along with B cell receptors and the B cells' idiotypic immune network. The B cell word agents are then uniformly distributed into the grid. In the learning stage, antigen

word agents are constructed one by one from a sentence from the dependency Treebank and are injected into the immune environment. According to the principles of clonal selection and the idiotypic immune network, B cells and antigens interact with each other, resulting in higher strength between B cells. In the last evaluation stage, the sentences in the test set are structured as dependency trees based on the model using a maximum spanning tree algorithm and are evaluated by computing unlabeled attachment scores (UASs) [8], i.e., the percentage of words that have the correct heads. The pseudo code of the MWAALM model is given in Algorithm 1.

Algorithm 1 Pseudo code of the MWAALM model.

Initialization:
1: Initialize immune environment as an $M \times M$ grid.
2: Initialize BWAs from the training set.
Learning:
3: Do for each sentence in the training set.
3.1: Construct AgWAs from the training sentence and inject them into the grid.
3.2: BWAs and AgWAs move freely until BWAs can recognize AgWAs.
3.3: BWAs make clones.
3.4: Clones undergo hypermutation.
3.5: Mutated clones are evaluated by a fitness function and the best fitting one is reserved.
3.6: The reserved BWAs act as antigens in the artificial immune network following spreading activation.
Evaluation:
4: Sentences in the test set are structured as dependency trees based on the model.
5: Evaluate the model by computing UAS of the predicted dependency trees.

2.2. Multi-word-agent Autonomous Learning Model

The proposed model, MWAALM, is constructed as an AOC system and is described following the formal and common framework of AOC systems. The AOC system contains a group of autonomous word agents and an environment where agents reside. We formally define the model as follows:

Definition 1 (Multi-word-agent Autonomous Learning Model): The Multi-word-agent Autonomous Learning Model (MWAALM) is a tuple $<\{w_1, w_2, \cdots, w_i, \cdots, w_N\}, E, \Phi>$, where $\{w_1, w_2, \cdots, w_i, \cdots, w_N\}$ is a group of autonomous word agents, E is an environment in which agents reside, and Φ is a system objective function guiding the model to evolve toward certain desired states.

2.3. Environment

As formulated by Jiming Liu, the environment, one of the main components in an AOC system, plays three roles [9]. The environment of the proposed AOC model functions according to these three roles. First, the environment serves as a work space in which word agents act and interact. Second, the environment can also act as a ‘notice board’ where word agents can post and read their sharable information such as word property values. Third, the environment maintains a central clock that helps synchronize the behaviors of all word agents, such as moving, clone, and mutation. The environment is formally defined as follows:

Definition 2 (Environment): The environment E in the MWAALM is an $M \times M$ grid and is characterized by an attribute set $\varepsilon s = \{es_1, es_2, \cdots, es_{N_{es}}\}$, where each es_i corresponds to a unique word property and N_{es} denotes the number of all unique word properties.

In the environment, each site is surrounded by eight adjacent sites, shown as Figure 3. More than one agent can reside in a site and can move to one of the adjacent sites freely and randomly. Each es_i corresponds to a unique word property; therefore, εs is a shared 'notice board' by which word agents can share their information. At each moment, εs represents the current state of the environment, composed of the current state of each word agent. At the end of learning, εs also represents the learning result of the model.

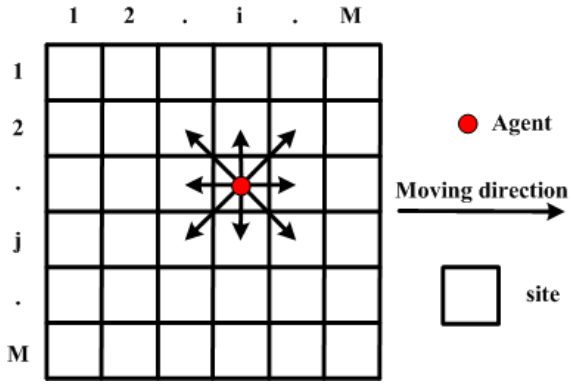

Figure 3. The environment in the MWAALM is an $M \times M$ grid in which agents reside.

2.4. Word agents

The basic elements of the model are word agents. Each word agent acts autonomously to achieve its goal, which is to improve the strength between it and its matched word agents. To achieve their respective goals, word agents perform their primitive behaviors with respect to the evaluation results of their states and comply with their behavioral rules. Through interactions, word agents can self-organize to achieve the system goal of optimizing the relation strength between words. We define a word agent as follows:

Definition 3 (Word Agent): A word agent w is a tuple $< S, F, G, B, R >$, where S denotes the current state of w, F is an evaluation function, G is the goal set of w, and B and R are primitive behaviors and behavior rules, respectively.

A word agent can only interact with adjacent agents, namely, neighbors. Before continuing this description, the neighbors of a word agent are defined in advance:

Definition 4 (Neighbors): The neighbors of a word agent w are a group of word agents $L^w = \{l_1^w, l_2^w, \cdots, l_i^w, \cdots, l_{N_L^w}^w\}$, where l_i^w resides in the same site of the grid as w does and N_L^w is the number of neighbors.

In the proposed model, there are two types of word agents: antigen word agents (AgWAs) and B cell word agents (BWAs). AgWAs simulate antigens, and BWAs

simulate B cells. In the following, the state, evaluation function, goal, primitive behaviors and behavior rules of AgWAs and BWAs are described.

2.4.1 Representation of BWA: All words in the annotated sentences of the training set are used to construct BWAs and word properties extracted from head-dependent pairs are used to construct BWAs' receptors. To initialize mimic idiotypic immune network of BWAs, each BWA holds a head-word set, containing all its head words in the training set, and a dependent-word set, containing all its dependent words in the training set.

State: The states of a BWA include living states and properties.

At each moment, the living state of a BWA may be one of the following four states: ACTIVE, STIMULATED, CLONAL, and MUTATED. The living state is the result of behaviors and is involved in behavior rules. When an agent first enters the AIS system or after affinity maturation, the agent is ACTIVE. Once the agent is stimulated by an antigen, the agent becomes STIMULATED. The stimulated agent makes clones, and each clone remains CLONAL. Then, each clone undergoes hypermutation, resulting in a mutated receptor and the MUTATED state. The mutated agent can stimulate other agents by spreading activation before finally returning to the ACTIVE state. The living state transition of a B cell word agent is shown in Figure 4.

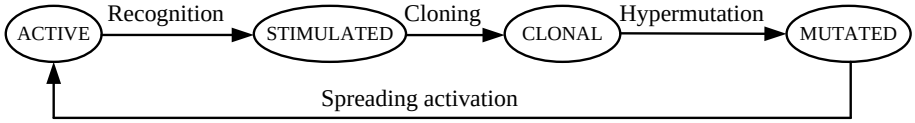

Figure 4. Transition of living states of a BWA

The properties of BWAs are composed of features of head-dependent pairs extracted from dependency trees of the training set and are grouped into head properties and dependent properties. In this proposed model, B cells represent words in the training set, and paratopes and idiotopes on the receptors of B cells represent the head properties and dependent properties of words, respectively. Dependency features extracted from the head-dependent pairs of the training set, according to the feature templates shown in Table 2, are used as properties of words. For a word w, $\{hf_1^w, hf_2^w, \cdots, hf_i^w, \cdots hf_{N_{hf}}^w\}$ is the head feature set of w extracted from all head-dependent pairs, whereby w is the head word and ω_i^w is the weight of hf_i^w. $\{df_1^w, df_2^w, \cdots, df_j^w, \cdots df_{N_{df}}^w\}$ is the dependent feature set of w and is composed of features extracted from all head-dependent pairs in which w is the dependent word. $P_{h-1}\_P_h\_P_{d-1}\_P_d$

Table 2. Feature templates of dependency

Word(W)	POS(P)	Word and POS
W_h	$P_h, P_d, P_h\_P_d$	$W_h\_W_d\_P_d$
W_d	$P_h\_P_{h+1}\_P_{d-1}\_P_d$	$W_h\_P_h\_W_d$
$W_h\_W_d$	$P_{h-1}\_P_h\_P_{d-1}\_P_d$	$W_h\_P_h\_P_d$
	$P_h\_P_{h+1}\_P_d\_P_{d+1}$	$P_h\_W_d\_P_d$
	$P_{h-1}\_P_h\_P_d\_P_{d+1}$	$W_h\_P_h\_W_d\_P_d$

In Table 2, given a head-dependent pair, W_h denotes the head word, W_d denotes the dependent word, P_h denotes the POS of the head word, P_d denotes the POS of the dependent word, P_{h+1} denotes the right adjacent word of the head word, P_{h-1} denotes the left adjacent word of the head word, P_{d+1} denotes the POS of the right adjacent word of the dependent word, and P_{d-1} denotes the POS of the left adjacent word of the dependent word. For example, *浦东(Pudong)←建设(construction)* is a head-dependent pair in the dependency tree shown in Figure 1, and NR and NN are their corresponding POS tagged below them. Then, features of the head-dependent pair include *浦东, 建设, 浦东_建设, NR, NN, NR_NN*, etc.

Given word properties extracted from head-dependent pairs, the paratopes P^w and idiotopes I^w of a BWA w are formulated as Equation (1) and (2).

$$P^w = \{(hf_1^w, \omega_1^w), (hf_2^w, \omega_2^w), \cdots, (hf_i^w, \omega_i^w), \cdots (hf_{N_{hf}}^w, \omega_{N_{hf}}^w)\} \quad (1)$$

$$I^w = \{df_1^w, df_2^w, \cdots, df_j^w, \cdots df_{N_{df}}^w\} \quad (2)$$

Evaluation function: A BWA assesses its states using evaluation functions and determines locally which agent should be recognized. Recognition between two agents is determined by their affinity; therefore, evaluation functions are defined as affinity functions shown as Equation (3) or (4). In this model, a BWA w_B can be stimulated by an AgWA w_{Ag} or an antigen-like BWA $w_{B'}$, i.e., a stimulated BWA, if w_{Ag} or $w_{B'}$ is a dependent word of w_B. If there exists more than one agent with which the w_B can be matched with in the local site, the agent with the maximum affinity between them is selected to be matched.

$$f_{affinity}(w_B, w_{Ag}) = \sum_{i=1}^{N_{hf}^{w_B}} \sum_{j=1}^{N_{df}^{w_{Ag}}} \delta\left(hf_i^{w_B}, df_j^{w_{Ag}}\right) \omega_i^{w_B} \quad (3)$$

$$f_{affinity}(w_B, w_{B'}) = \sum_{i=1}^{N_{hf}^{w_B}} \sum_{j=1}^{N_{df}^{w_{B'}}} \delta\left(hf_i^{w_B}, df_j^{w_{B'}}\right) \omega_i^{w_B} \quad (4)$$

$$\delta(x, y) = \begin{cases} 1, & if\ x = y \\ 0, & otherwise \end{cases} \quad (5)$$

Goal: The goal of a BWA is to regulate the affinity between itself and another agent. To achieve this goal, the BWA obtains some increments for the properties' weight when hypermutation occurs at the receptor of the BWA.

Behavior: A BWA has five primitive behaviors: moving, recognition, spreading-activation, cloning and hypermutation.

- Moving: The BWA can move randomly to adjacent sites or stay where it resides.
- Recognition: A BWA recognizes other neighbor AgWAs with its paratopes according to the affinity between them. The BWA is stimulated and assigned an initial integer activation value as the activation level $L_{activation}$.

- Spreading-activation: A stimulated BWA w can act like an antigen and transfer its activation value to another BWA, which is a head word of w, with the activation value reduced by one. The process of activation propagation continues until the activation value decreases to zero. The initial activation level $L_{activation}$ determines the spreading depth in the immune network. If $L_{activation}$ is set to zero, the model simply reduces to a clonal selection algorithm.
- Cloning: Once a BWA w is stimulated by another agent, it reproduces a group of clones $\{w'_1, w'_2, \cdots, w'_i, \cdots, w'_K\}$, where K is the number of clones.
- Hypermutation: Each clone w' of the BWA w suffers hypermutation individually. In hypermutation, the weight $\omega_i^{w'}$ of each paratope of the agent's receptor is assigned a random increment $\Delta_i^{w'}$ with a certain probability $p_{mutation}$. $\Delta_i^{w'}$ is inversely proportional to the fitness of the agent and to the affinity between the agent and the recognized antigen. The mutation is performed according to Equation (6):

$$\omega_i'^{w'} = \omega_i^{w'} + \Delta_i^{w'},$$

$$\Delta_i^{w'} = \alpha * (1/\beta) * N(0,1),$$

$$\alpha = \exp(-f_{affinity}) * \exp(-f_{fitness}(w')), \quad (6)$$

where $\omega_i'^{w'}$ is the mutated weight, $N(0,1)$ is a Gaussian random variable of zero mean and standard deviation $\sigma = 1$, β is a parameter that controls the decay of the inverse exponential function, $f_{affinity}$ is the affinity determined by Equation (4) or (5), and $f_{fitness}(w')$ is the fitness of each clone determined by a fitness function that will be introduced later. These clones will be evaluated by a fitness function, and the best fitting will be reserved and replace its parent. In the model, the initial value of ω_i^w is set to zero.

Behavior rules: A BWA has two behavior rules: the moving rule and the mutating rule.

- Moving rule: The agent can randomly move to one of eight adjacent regions or not at all. Thus, the agent chooses one direction to move with a probability of 1/9.
- Mutating rule: Each paratope of the receptor of the B cell undergoes mutation with a certain probability $p_{mutation}$.

2.4.2. Representation of AgWA: In each round of learning, one sentence dependency tree is picked from the training set. The dependent word of each head-dependent pair of the dependency tree is used to construct an AgWA, and features of the head-dependent pair are used as properties of the AgWA.

State: The states of AgWAs include living states, lifetimes and properties.

The living state of an AgWA may be ACTIVE or DEAD. When an antigen first enters the AIS system, the agent is ACTIVE and is assigned an initial lifetime value. Once the agent is recognized by a B cell, the agent becomes DEAD and will be cleaned. The lifetime $L_{lifetime}$ of an AgWA, an integer value, represents the survival time of an agent. If the lifetime decreases to zero, namely via aging, the agent also becomes DEAD. The transition of AgWA living states is shown in Figure 5.

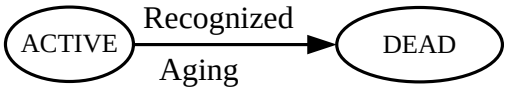

Figure 5. Transition of living states of an AgWA

The properties of an AgWA are composed of features of its head-dependent pair and are used as epitopes of the antigen. The epitopes E^w of an antigen word agent w are formulated using Equation (7).

$$E^w = \{df_1^w, df_2^w, \cdots, df_i^w, \cdots df_{N_{df}^w}^w\} \quad (7)$$

Evaluation function: An AgWA is always passively recognized by other B cell words. An ACTIVE AgWA continues moving until it is recognized by a BWA and then becomes DEAD. Thus, no evaluation function is designed for an AgWA.

Goal: The goal of an AgWA is to be recognized.

Behavior: The only behavior of an AgWA is to move randomly to an adjacent region or stay where it resides. When the antigen moves a step, its lifetime reduces by one. The antigen keeps moving until it is recognized or until its lifetime decreases to zero.

Behavior rules: The agent moves following the same moving rule as a BWA.

2.5. System Objective Function

The system objective function of this model is a global measurement for the performance of word strength regulation. It guides the model to evolve toward the desired states in that word strength are well tuned.

When a clone w' of the BWA w finishes its hypermutation, the weight of the paratopes of its receptor may be changed, and the word strength may be regulated. If the words strength are well tuned, then the training sentence can be transformed into a correct dependency tree using word strength in a bottom-up manner[8], [10]. The system objective function Φ of the model is designed as a measurement function for the goodness of the predicted dependency tree of the training sentence from which antigens are built. According to the function, the environment decides whether the regulations are accepted or rejected. If the regulations are accepted, the B cell clone will be reserved and replace its parent or the B cell clone will be eliminated from the environment. Thus, the function Φ is also a fitness function for w'.

The dependency tree of a sentence is identical to its maximum spanning tree, in which nodes are words and edges are dependency relations between words[8]. The strength between any two words can be determined according to equation (4). The dependency tree of a sentence, which is produced using words and word strength, can

be predicted using a maximum spanning tree algorithm such as the Chu-Liu-Edmonds algorithm[11], [12] and Eisner's algorithm[10]. For Chinese sentences, dependency trees are projective; therefore, Eisner's algorithm is used as the maximum spanning tree algorithm. The goodness of a predicted dependency tree can be measured based on two aspects. On the one hand, the percentage of words that have the correct predicted heads, denoted as f_{UAS}, directly indicates the precision of the predicted dependency tree. On the other hand, the annotated dependency tree of a sentence should theoretically be the maximum spanning tree, which means that the score of the annotated dependency tree (i.e., the sum of the strength of the head-dependent pair) should be higher than the score of any other spanning tree. Therefore, the difference between the score of the annotated dependency tree and that of the predicted dependency tree, denoted as f_{score}, can indirectly indicate the goodness of the predicted dependency tree.

Let S be a training sentence, w_i^S be a word in S, $T=(V,E)$ be the annotated dependency tree, and $T'=(V,E')$ be the predicted dependency tree based on the state values of V. Let $V=\{w_1^S, w_2^S, \cdots, w_i^S, \cdots, w_{N_S}^S\}$ be the node set of tree T or T', where N_S is the number of words in the sentence S and E and E' are the edge sets of T and T', respectively. f_{UAS} and f_{score} are formulated using Equation (8) and (9). The system objective function Φ combines f_{UAS} and f_{score} and is defined using Equation (11).

$$f_{UAS}(T',T)=\frac{|E'\cap E|}{|E|} \quad (8)$$

$$f_{score}(T',T)=\frac{score(T)}{score(T')} \quad (9)$$

$$score(T)=\sum_{e\in E} score(e)=\sum_{e\in E} f_{affinity}(w_{head}^e, w_{dependent}^e) \quad (10)$$

$$\Phi(w_1^S, w_2^S, \cdots, w_i^S, \cdots, w_{N_S}^S)= f_{UAS}(T',T) * f_{score}(T',T)=\frac{|E'\cap E|}{|E|} * \frac{score(T)}{score(T')} \quad (11),$$

where $f_{UAS}(T,T')$ is the UAS of T', i.e., the percentage of words that have the correct predicted heads, and $score(T)$ is the score of a dependency tree and is defined as the sum of the score of all edges in the tree. The system objective function Φ is also used as the fitness function of a B cell clone in the context of the training sentence. For the B cell clone w', the fitness function is defined in Equation (12).

$$f_{fitness}(w')=\Phi(w_1^S, w_2^S, \cdots, w', \cdots, w_{N_S}^S) \quad (12)$$

According to Equation (12), the best clone w'^* is determined from the group of clones $\{w_1', w_2', \cdots, w_i', \cdots, w_K'\}$ of w. The clone w'^*, which has a maximum fitness value, may be reserved and may replace its parent while the others are eliminated.

$$w'^*=\arg\max_i(f_{fitness}(w_i')) \quad (13)$$

If $f_{fitness}(w'^{*}) > f_{fitness}(w)$, then w is replaced by w'^{*}; otherwise, w is still replaced by w'^{*} but with probability $p_{reserve}$. The fitness function of this model is a global measurement for the performance of word strength regulation, which guides the model to evolve toward the desired state, in which combination strength between words are well tuned.

2.6. Summary of the model

As described above, MWAALM includes three components: the environment, word agents and the system objective function. The environment is an $M \times M$ grid and characterized by a group of shared attribute set εs; word agents, including BWAs and AgWAs, are basic components of the model and are built from words in the training set; the system objective function Φ, also used as the fitness function $f_{fitness}$ of an agent's clone, is a global measurement for the performance of word strength regulation. Elements of the word agents are summarized in Table 3.

Table 3 Summary of elements of word agents.

Elements of word agents	**BWAs**	**AgWAs**
States	living state and properties	living state and properties
Evaluation functions	$f_{affinity}$	/
Goal	Affinity regulation	be recognized
Behaviors	Moving, Recognition, Spreading-activation, Cloning, Hypermutation	Moving
Behavior rules	Moving rule and mutating rule	Moving rule

There are several parameters in the model: some for agents and others for the environment. The parameters and their default values are listed in Table 4. This model is based on the clonal selection principle and immune network theory. Word agents can spread activation to other word agents through the immune network. The initial activation level $L_{activation}$ determines the spreading depth in the immune network. If $L_{activation}$ is set to zero, the model simply reduces to a clonal selection algorithm. Thus, $L_{activation}$ is the most important parameter in the model. During the experiments, $L_{activation}$ is set to different values, i.e., 0 and 3, to investigate the impact of the immune network or the language network on the model. The other parameters are set to their default values.

Table 4. List of parameters in the model

Parameters type	Parameters	Description	values
Agent parameters	$L_{lifetime}$	The lifetime of an antigen	3
	$L_{activation}$	The activation level of an activated B cell word agent	{0, 3}
	K	The number of B cell clones	5
	$p_{mutation}$	The probability with which a B cell word agent mutates at its paratopes	0.5
	β	The parameter that controls the decay of the inverse exponential function	100
Environment parameters	M	The row number of the $M \times M$ grid	30
	$p_{reserve}$	The probability with which the best-fitting B cell word agent is reserved	0.5

References

[1] J. Kuby, T. J. Kindt, B. A. Osborne, and R. A. Goldsby, Kuby Immunology, Fifth. New York: W.H. Freeman and Company, 2002.

[2] E. Hart, H. Bersini, and F. Santos, "Structure versus function: a topological perspective on immune networks.," Nat. Comput., vol. 9, no. 3, pp. 603–624, 2009.

[3] E. Hart, "Analysis of a Growth Model for Idiotypic Networks," in Artificial Immune Systems SE - 6, vol. 4163, H. Bersini and J. Carneiro, Eds. Springer Berlin Heidelberg, 2006, pp. 66–80.

[4] R. Hudson, An Introduction to Word Grammar. Cambridge University Press, 2010.

[5] A. S. Perelson, "Immune network theory.," Immunol. Rev., vol. 110, pp. 5–36, Aug. 1989.

[6] A. M. Collins and E. F. Loftus, "A spreading-activation theory of semantic processing.," Psychol. Rev., vol. 82, no. 6, pp. 407–428, 1975.

[7] L. N. De Castro and J. Timmis, Artificial Immune Systems: A New Computational Intelligence Approach. Springer, 2002, p. 380.

[8] S. Kubler, R. McDonald, and J. Nivre, "Dependency Parsing," Synth. Lect. Hum. Lang. Technol., vol. 2, no. 1, pp. 1–127, Jan. 2009.

[9] K. C. T. Jiming Liu, XiaoLong Jin, Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling. Springer, 2005.

[10] J. M. Eisner, "Three new probabilistic models for dependency parsing: An exploration," Proc. 16th Conf. Comput. Linguist. 1, vol. 96, no. August, pp. 340–345, 1996.

[11] T.-H. L. Chu, Yoeng-Jin, "On the shortest arborescence of a directed graph," Sci. Sin., vol. 14, pp. 1396–1400, 1965.

[12] J. Edmonds, “Optimum branchings,” J. Res. Natl. Bur. Stand. (1934)., vol. 71B, pp. 233–240, 1967.