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

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

Abstract

Words are basic structural units of language that combine with each other to form sentences. The learning strength of combinative relations between words is of key importance in sentence structure analysis. Inspired by the analogies between words and lymphocytes, a multi-word-agent autonomous learning model based on an artificial immune system is proposed to learn word combination strength. The model is constructed via Cellular Automation, and words are modeled as B cell word agents and as antigen word agents. The combination strength between words is viewed as affinity of specific recognition relations between B cells. The language network is then simulated as an immune network. Meanwhile, Spreading Activation, which is the memory mechanism by which humans uses language networks in cognitive psychology, is employed to simulate idiotypic interactions between B cells. This research provides a completely new perspective on language and words and introduces biologically inspired processes from immune systems into the proposed model. The most significant advantage of the model is the ability of continuous learning and the concise implementation method. According to the graph-based dependency parsing method, the syntax dependency tree of a sentence can be predicted based on word combination strength in a bottom-up paradigm, from pairs of smaller structures to larger structures. Therefore, the effectiveness of the model can be verified by sentence dependency parsing. The experimental results on the Penn Chinese Treebank 5.1 indicate that our model can effectively and continuously learn word combination strength.

Keywords: *word agent, word combination strength, artificial immune system, language network, spreading activation*

1. Introduction

Words are the basic structural unit of language. Words interact with each other and follow certain rules to form sentences. As Saussure states, 'Language is a system of inter-dependent terms in which the value of each term results solely from the simultaneous presence of the others' [1]. In these interactions, some words depend on or are depended upon by others. These dependencies can be represented as a word network, also called a language network[2]. As an example, figure 1 shows a Chinese sentence "上海浦东开发与法制建设同步 (Development is synchronized with legal construction in Pudong of Shanghai)", which is an excerpt from the Penn Chinese Treebank 5.1 (CTB)[3] and is annotated to a dependency tree. In the dependency tree, each dependency relation holds between a syntactically subordinate word, called the

dependent, and another word called the head, upon which the subordinate word depends on. Dependency relations are also called head-dependent pairs and are represented by arrows pointing from the head to the dependent. Consequently, a language network can be constructed with head-dependent pairs from a large number of dependency trees. One of the attractive points of the network perspective on language is the possibility of analyzing language in the same way as other types of knowledge. Everything in a language can be formally described in terms of nodes and edges between nodes[2], that is, words and their inter-dependent relations. Relations between different words may exhibit different strength. Combination strength measures the degree of affinity of a combinative relation, where a higher strength between words indicates that they prefer to combine with each other. The combination strength, determined by the features of the context of the combinative relations, are the rules that determine the order by which words compose a sentence. Based on the combination strength between two words in a sentence, the sentence structure, syntax structure or semantic structure, may be created in a bottom-up paradigm, from pairs of smaller structures to a larger structure[4]. With this understanding, regulating or learning the combination strength between words is of key importance in sentence structure analysis.

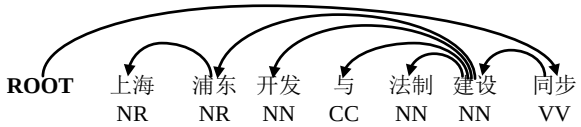


Figure 1. The dependency structure for a Chinese sentence" 上海浦东开发与法制建设同步 (Development is synchronized with legal construction in Pudong of Shanghai)".

This research focuses on words and aims to learn the combination strength between them. This type of learning, that is, learning to combine words to form grammatical sentences, also occurs in human language development. Human language development is a life-long process[5]. Analogously, our body's immune system uses a computational strategy to perform its many functions in terms of protecting and maintaining the body, and it continuously develops this strategy based on experience[6]. When pathogens, also called antigens (Ags), invade the human body, bone cells (B cells), the basic components of the immune system, recognize antigens by their receptors. The B cells then undergo a sequence of state changes, resulting in higher affinities between the B cells and antigens, finally secreting a large amount of antibodies, which can kill the invading antigens. The immune system is also considered to be a continuous learning system because of its ability to adapt to foreign antigens[7]. Another important analogy between language and immune systems was first made in Jerne's Nobel lecture, in which the variable region of a given antibody molecule was interpreted as a sentence or a phrase. Inspired by Jerne's lecture, we make an analogous comparison between B cells and words: B cells can only recognize some types of antigens with specific receptors on the surface, and words can only combine with certain other words with properties such as word tokens, word classes (i.e., POS), meaning, and valency [2]. The similar learning characteristics and analogies between words and B cells inspire our research to learn from the immune system. The human immune system has attracted greater attention as an evolutionary model and has inspired research of artificial immune

systems (AISs). There are four fundamental types of AIS models: negative selection mechanisms[8], clonal selection theory[9], idiotypic immune networks[10], and dendritic cell algorithms[11].

This research presents a multi-word-agent autonomous learning model (MWAALM) based on an AIS using a clonal selection algorithm and an idiotypic immune network to regulate the strength of combinative relations between words. First, words are viewed as B cells and antigens and are modeled as B cell word agents (BWA) and antigen word agents (AgWA); word combinative relations are viewed as recognitions between B cells. The word combination strength are represented by the affinities between B cells and are regulated by applying a clonal selection algorithm and an idiotypic immune network. Considering the sentence dependency tree bank to be available, we narrow the combinative relation to a syntax dependency relation. This model is evaluated by a graph-based dependency parsing method using a maximum spanning tree algorithm[4]. Second, spreading activation[12] is introduced to simulate the dynamics of the idiotypic immune network. In our model, B cells are words; therefore, the idiotypic immune network is equivalent to a language network. The spreading activation is the memory mechanism of humans using a language network[2]. Naturally, the spreading activation is employed as the idiotypic response mechanism, and the activation level is employed as the idiotypic level. Third, cellular automation (CA) [13] in the autonomy oriented computing (AOC)[14] framework is employed to construct the model. The immune system is composed of autonomous lymphocytes, i.e., a type of immune system cell[15]. As an agent-based modeling method, CA is a natural application for modeling cellular systems[16]. Being a generic framework, AOC offers a new computing paradigm that makes use of autonomous entities for solving computational problems and modeling complex systems[17]. This paper presents a completely new perspective on language and proposes an autonomous learning model to learn combination strength between words. The most significant advantages of the proposed model are its ability to continuously learn and its concise implementation methodology. Since well-tuned word combination strength can induce the sentence syntax dependency tree bottom-up, this model is indirectly validated via sentence dependency parsing. The experimental results based on CTB indicate that this model can effectively and continuously learn word combination strength.

The remainder of this paper is organized as follows. In Section 2, the biological inspirations from immune systems are discussed. Related work is summarized in Section 3. An autonomous learning model based on adaptive immune theories and spreading activation is proposed in detail in Section 4. In Section 5, the experimental results of the model are presented and analyzed. Finally, the conclusion of the work is given in the last section.

2. 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[18], 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 2 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.

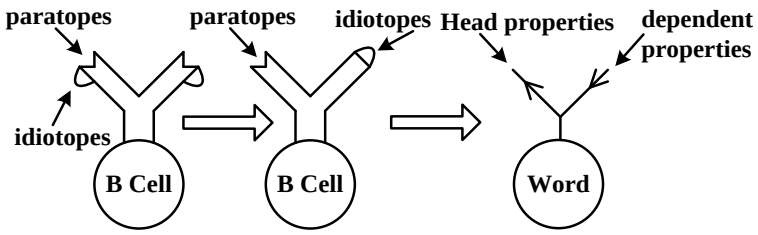


Figure 2. The design of B cell receptors and word properties

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 [19]. According to the simulation, a power-law degree-distribution can emerge[20]. Similar to an immune network in terms of complexity, the language network is also a complex network[2]. 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[21]. Spreading activation is used in cognitive psychology to model the

fan-out effect[12] and is also used as the memory mechanism of humans using a language network[2]. In this AIS model, spreading activation is employed as the idiotypic interaction mechanism as follows. As shown in Figure 3, 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.

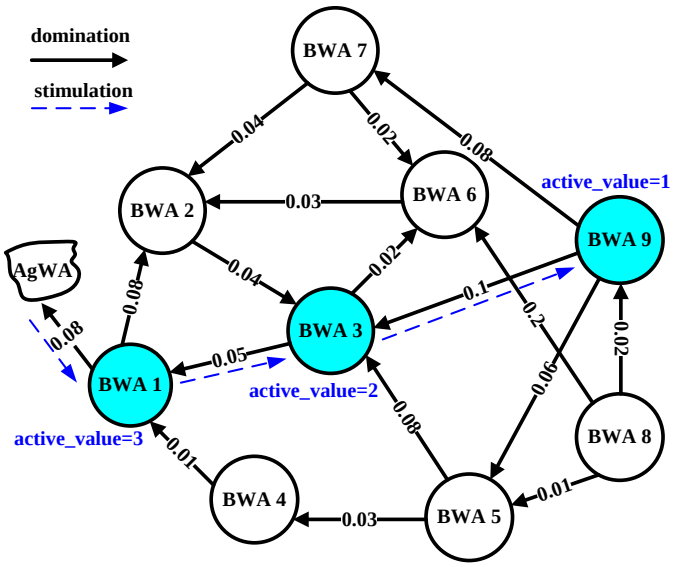


Figure 3. The process of spreading activation in the artificial immune network of this model

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[22]).

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

3. Related work

Analogies between words and lymphocytes were first proposed by Dong[23]. Inspired by Dong's work, Yang[24] developed a revised model and proposed a lymphocyte-style word representation[25]. However, both the two models did not employ the immune network theory comprehensively. As a major extension of Yang[24], this research introduces the immune network theory into the MWAALM and redesigns the hypermutation behavior and the system objective function. Related work mainly involves studies on AISs. AISs have been commonly defined, as in[22], as "Adaptive systems, inspired by theoretical immunology and observed immune functions, principles and models, which are applied to problem solving." From this definition, it can be deduced that an AIS has to satisfy four minimal requirements: it must incorporate basic immune components, model immune functions and principles, solve problems, and have adaptive capability. Research on AISs can be divided into immune modeling, theoretical AISs and applied AISs[26]. Research on applied AISs is the most vibrant in the field and can be summarized as (1) learning, (2) anomaly detection and (3)

optimization[27]. Generally, learning can be understood as the process of acquiring knowledge from experience and being able to re-apply that knowledge to previously unseen problem instances. This research simulates lymphocytes as autonomous word agents and employs immune system principles to learn and regulate combination strengthens between words according to the annotated head-dependency pair. Thus, this research represents immune-based learning of applied AISs. Related works, including immune-based learning and agent-based modeling, are discussed in the remainder of this section.

3.1. Immune-based learning

Dasgupta originally defined the seminal list of features of an AIS, which includes feature extraction, recognition and learning[28]. These features are also key features of classical machine-learning algorithms. Because of these commonalities, immune-based learning has gained the most attention in the area of applied AIS and has been applied in clustering, classification and pattern recognition, robotics and control applications, among others. To date, there have been four types of AIS algorithms used in applied AIS[26]: negative selection algorithms, clonal selection algorithms, immune network algorithms and dendritic cell algorithms. Immune-based learning mainly involves clonal selection algorithms and immune network algorithms.

Clonal selection theory describes the basic feature of adaptive immune response: only those B cells that recognize antigens proliferate, and the offspring may undergo somatic hypermutation, resulting in a higher affinity with antigens[9]. According to clonal selection theory, the process of clonal selection comprises four elementary stages: recognition, clonal expansion, hypermutation and a selection mechanism. Recognition triggers the process, clonal expansion generates candidates, hypermutation enables the leveraging of candidates, and selection is employed to reserve the fittest candidates. Clonal selection theory has inspired the unsupervised learning model CLONALG[29] and the supervised AIS classifier AIRS[30]. A new multiclass classifier based on the clonal selection principle, whose unique feature is the embedded property of the local feature selection, has also been proposed[31].

Idiotypic immune network theory was first proposed by Jerne[10] and formalized into a model by Farmer[7]. In this theory, B cells can recognize or be recognized by other B cells until the idiotypic level is maximized, which leads to the creation of a network among B cells[10], [21]. By employing the metaphor of immune network theory, unsupervised learning models were also proposed in[32]. After Hunt and Cooke first attempted to introduce immune network algorithms into a supervised binary classifier[33], many AIS classifiers were proposed[34], [35].

In the era of big data, huge amounts of data are collected and stored cheaply and easily. A more promising application area for AIS may be dynamic clustering or classification[27]. It has been proposed that this type of AIS incorporate some form of memory to balance the need to maintain a record of currently underutilized knowledge acquired in the past against the need to store newly acquired knowledge that is valuable in the current climate. For more AIS applications, please refer to[26], [27], [36].

3.2. Agent-based modeling

In artificial immune system modeling, there are two types of competing modeling approaches: continuous modeling and discrete modeling[37]. A continuous model is generally described by a set of differential equations that yield average behaviors and results for the total population. This approach is not very intuitive, and it ignores important aspects of the

immune response, such as the locality of responses and the diversity of repertoires. A discrete model can be implemented as a multi-agent system via agent-based modeling, which employs large numbers of autonomous agents that interact with each other in an artificial environment. The agents' behaviors are described by rules that determine how they learn, interact and adapt. The agents and the environment are generally implemented with a cellular automaton (CA). Agent-based and CA approaches are generally well suited for modeling complex systems and IS in particular, providing a way to represent the true diversity of IS entities and offer modeling possibilities close to biological reality[38]. One of the most referenced and peer reviewed IS simulators, ImmSim, was based on CA with probabilistic rules[39]. At each time step, cellular entities in the same CA site can interact with each other stochastically and diffuse through the lattice. C-ImmSim is a version of ImmSim developed by F. Castiglione and M. Bernaschi in the C programming language, with a focus on improved efficiency and simulation size and complexity[40]. C-ImmSim is the most advanced IS simulator based on the original version. Considering that there is few research on the development process of simulation models for the immune system, a descriptive guide was introduced for the development of a simulation model in immunology and the challenges that might be encountered during this process, with such guidance applicable to all simulation methods in immunology[41].

In agent-based and CA models, large numbers of agents interact autonomously with each other and with the environment according to their states and behavior rules. Autonomy Oriented Computing (AOC) is a generic and formulated framework for modeling multi-agent systems[14]. Under the AOC framework, each agent is defined as a tuple comprising the state, evaluation function, goal, behaviors and behavior rules, and the environment is defined as an infinite space wherein agents reside and is characterized by a set of states. The formulated definitions of agents and the environment provide not only specifications but also guidelines for modeling agents and the environment. A system objective function is also defined in the AOC framework, which guides the multi-agent system to evolve from unorganized to organized and from a bad organization to a good one, namely, self-organization[42]. AOC has been applied to complex system modeling applications such as extracting web user behaviors [43], analyzing the dynamics of social networks[44], and solving complex problems such as distributed optimization[45], [46] and mining network communities[47], [48]. The AIS model of this research follows the idea of C-ImmSim and is developed under the AOC framework.

Although there have been many successful applications of AIS, Emma Hart argued that there are no unique features of the problem domain that indicate that an AIS-based algorithm can offer anything above and beyond the more traditional machine learning algorithms[27]. Dipankar Dasgupta also proposed that distinctive immune-inspired algorithms could be developed without any logical or technique overlap for any existing techniques[26]. This research comprehensively exploits the consistency between the immune system and language, and it develops an autonomous learning model based on clonal selection and immune network theory as well as spreading activation. This research is expected to make positive contributions, as noted by Emma Hart and Dipankar Dasgupta.

4. Multi-word-agent autonomous learning model

4.1. 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) [49], 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.

4.2. 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.

4.3. Environment

As formulated by Jiming Liu, the environment, one of the main components in an AOC system, plays three roles [14]. 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 4. 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.

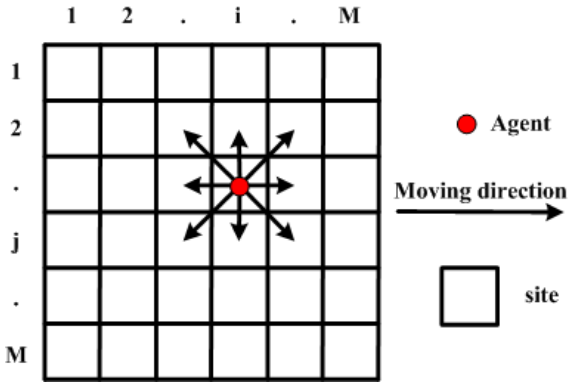


Figure 4. The environment in the MWAALM is an $M \times M$ grid in which agents reside.

4.4. 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.

4.4.1 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 5.

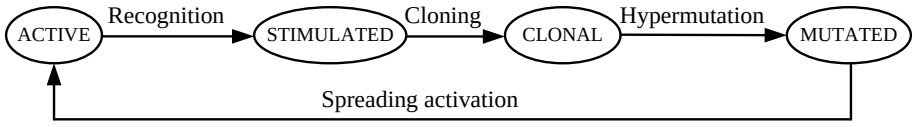


Figure 5. Transition of living states of a BWA

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 $p_{mutation}$.

4.4.2. 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 6.

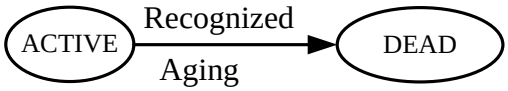


Figure 6. Transition of living states of an AgWA

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.

4.5. 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[4], [49]. 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[49]. 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[50], [51] and Eisner's algorithm[4]. 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.

4.6. 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

5. Experimental results

5.1. Data Sets and Experimental Design

The primary purpose of the experiments is to investigate the effectiveness of the regulation of word combination strength by the proposed model. According to the graph-based dependency parsing, well-tuned word combination strength can induce the sentence syntax dependency tree bottom-up. Thus, the model is indirectly validated by the task of sentence dependency parsing.

A dependency Treebank converted from the CTB is employed as experimental data. To construct the dependency Treebank, the Penn2Malt tool[52] and the head-finding rules[53] are used to perform the phrase-to-dependency conversion. The dependency Treebank is divided into a training set and a test set. All words of the sentences in the training set are used to initialize the BWAs, and the dependency relations between words were used to initialize the artificial immune network. Idiotopes and paratopes of the BWAs were equipped by features of head-dependent pairs. These features were extracted according to the feature templates shown in Table 2. The performance of the dependency parsing of the model is evaluated by the UAS on the test set, which is defined as equation (14).

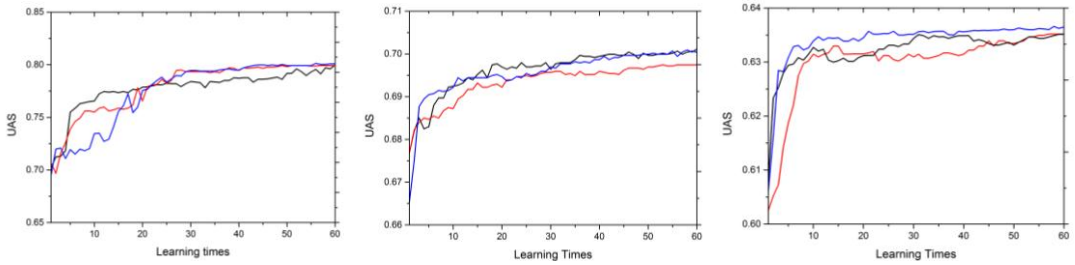
$$UAS = \frac{\#of\ \ words\,that\,are\,assigned\,correct\,heads\,in\,the\,test\ set}{\#of\ \ words\,in\,the\,test\ set} \quad (14)$$

It is well known that parsing systems tend to have lower accuracies for longer sentences. This is primarily due to the increased presence of complex syntactic constructions involving prepositions, conjunctions, and multi-clause sentences[54]. Furthermore, a graph-based parser's efficiency degrades dramatically when the input sentence becomes long. This is primarily due to the high time complexity of the utilized MST algorithm ($O(n^2)$ for the Chu-Liu-Edmonds algorithm and $O(n^3)$ for Eisner's algorithm). Thus, experiments are conducted on three data sets with different sentence lengths to verify the performance of the model. We define the length l of a sentence as the word count of the sentence. The sentence length l takes on three ranges: $l<10$, $10\le l<20$, and $20\le l<50$. Three groups of experiments are designed. The first group aims to validate the effectiveness of the model on the dependency parsing task with larger-scale data sets containing 1000 training sentences and 300 test sentences. Experiments on smaller-scale data sets may consume less time. Thus, the second group of experiments is designed to investigate the impact of $L_{activation}$ on the model on the smaller-scale data sets, which contain 100 training sentences and 50 test sentences. The third group of experiments is designed for comparison. Using the same data set and feature templates, the performance of the model is compared with the MSTParser[55], which is a graph-based parser that also uses the MST algorithm[4]. The model contains mutation and movement operators, and as such are likely stochastic in nature. Therefore experiments with multiple runs are performed to gain an accurate overview of results.

5.2. Results

The model is evaluated by computing UASs on a test data set when the model finishes a round of learning, and another round of learning follows. Curves with learning times on the x-axis and UASs on the y-axis are expected to continuously increase with increased learning time, eventually converging to a certain level.

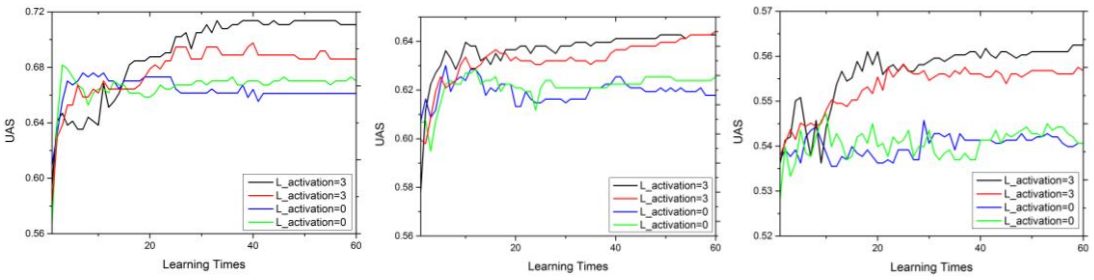
5.2.1 Performance of the MWAALM for dependency parsing: The experimental results on three larger-scale data sets with different sentence lengths are shown in Fig. 7. In these experiments, the parameters of the model are set to the same values, and $L_{activation}$ is set to 3. Although the three groups of result curves in Fig. 7 converge to different levels, they exhibit the same tendency. With continuous injection of antigen word agents, the precisions of the dependency parsing on the test sentences continuously increase, providing evidence that the proposed model can continuously and effectively learn and regulate relation strength between words.



(a) Results on the short-sentence ($l<10$) data set (b) Results on the medium-length-sentence ($10\le l<20$) data set (c) Results on the long-sentence ($20\le l<50$) data set

Figure 7. Results on three larger-scale data sets

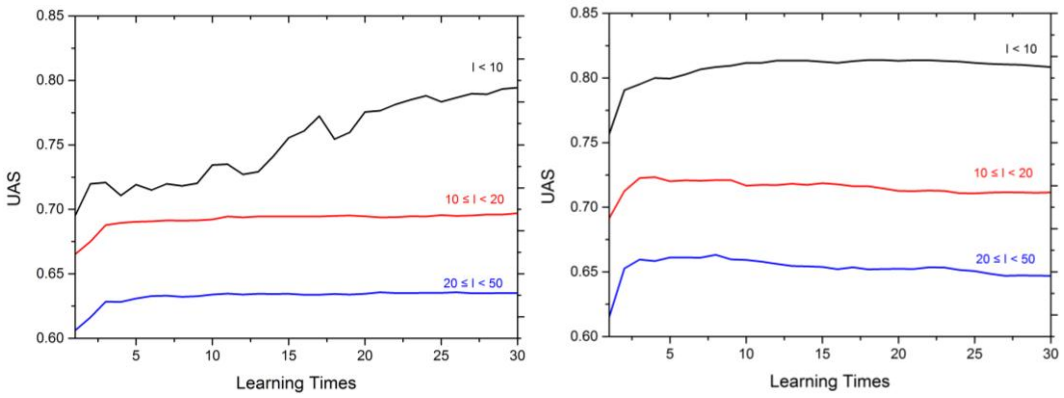
5.2.2 Investigation of $L_{activation}$ of MWAALM: The activation level $L_{activation}$ is an important parameter in MWAALM. Comparative experiments are conducted to investigate how this parameter affects the performance of MWAALM on three smaller-scale data sets with different sentence lengths. For comparison, on each data set, $L_{activation}$ is set to 0 and 3 in comparative experiments. Three groups of result curves are shown in Fig. 8. It is evident that curves with $L_{activation}$ = 0 fail to climb higher and even exhibit many fluctuations, as shown by sub-graph (c). Therefore, it is preferable to set $L_{activation}$ to 3 rather than to 0. In other words, spreading activation in the language network is an effective mechanism in the proposed model. In this mimic idiotypic immune network, a word agent may influence more word agents and naturally provide more chances for hypermutation for other agents. Moreover, the fitness of an agent can be evaluated on a broader scope.



(a) Results on the short-sentence ($l < 10$) datat set (b) Results on the medium-length-sentence ($10 \le l < 20$) datat set (c) Results on the long-sentence ($20 \le l < 50$) datat set

Figure 8. Comparative results with different activation levels on three smaller-scale data sets.

5.2.3. Comparisons between MWAALM and MSTParser: MSTParser operates primarily over arc scores, which are parameterized by a linear combination of a parameter vector and a corresponding feature vector for the arc. Under the generic online learning framework, MSTParser aims to learn the parameter vector: a single training sentence is considered in each iteration, and parameters are updated by applying an algorithm-specific update rule to the sentence under consideration. The main difference between MWAALM and MSTParser lies in how the parameter vector is learned. Therefore, comparisons between MWAALM and MSTParser are explored on the same data sets and for the same feature templates shown in table 2.



(a) Performances of MWAALM on different sentence-length data sets

(b) Performances of MSTParser on different sentence-length data sets

Figure 9. Comparisons of performances between MWAALM and MSTParser on different sentence-length data sets.

The two models are compared with different sentence-length data sets, as shown in Fig. 9. They present similar characteristics on the different data sets, namely, better performance with short-sentence data compared to long-sentence data and continuously improved performance with increased learning time, which is representative of the ability of continuous learning. However, note that the result curves of MWAALM continue increasing in the later stage, and the curves of MSTParser tend to decrease. The comparisons of the curves' trends demonstrate the potential of MWAALM.

The final results generated by MWAALM and MSTParser on three larger data sets are displayed in table 5. On the three different-sentence-length data sets, MWAALM performs approximately as well as MSTParser. As a completely new method to this classical NLP task, MWAALM's performance is comparable with MSTParser, which is a well-known graph-based dependency parser. Comparative results further indicate the effectiveness of the proposed model.

Table 5. Comparison of final results between MWAALM and MSTParser

Model	$l<10$	$10\le l<20$	$20\le l<50$
MSTParser	0.8093	0.7112	0.6471
MWAALM	0.8008	0.7012	0.6365

6. Conclusions and future works

This research presents a multi-word-agent autonomous learning model to regulate the combination strength between words based on adaptive immune theory and spreading activation. In this model, word agents locally determine their behaviors by themselves and do not explicitly know the global goal of the whole system, so behaviors of word agents and the goal-guided fitness function are designed concisely by applying the clonal selection mechanism and immune network theory. The model is evaluated on a dependency Treebank from the CTB.

The experimental results demonstrate that the model can continuously and effectively regulate word combination strength. The continuous learning characteristic is mainly attributed to the employment of the clonal selection mechanism and spreading activation in the language network. The effectiveness is mainly attributed to the global fitness function, which guides the model to evolve toward the desired state, in which the combination strength between words is well tuned.

With a concise and multi-agent modeling method, this AIS-based model obtains the ability to continuously learn, and it performs well for sentence dependency parsing, which is a classical research task in natural language processing (NLP). In the area of NLP research, applications of statistical machine learning methods are more prevalent. However, most statistical machine learning methods fail to adapt to new circumstances and lack the characteristic of continuous learning; this disadvantage greatly hampers both research on and the applications of NLP. The performance of this model may provide certain inspiration to studies on NLP as well as on machine learning.

Three aspects of future work will be focused on. In this research, words are viewed as lymphocytes and are represented as BWAs. This new lymphocyte-style representation is a two-vector word representation[56] and has the potential to express combinative relations, which is an inherent limitation of existing word representations such as distributed word representation[57]. Thus, a future goal is to investigate lymphocyte-style representations in various classical NLP tasks and perform comparisons with existing word representations. The semantic dependency relation between words is another type of combinative relation. According to the strength of semantic dependency relations, sentences can be parsed into semantic dependency trees. This research only involves the syntax dependency relation between words. Thus, another future goal is to adapt the model to regulate semantic relation strength between words. Compared with MSTParser, this model seems to be somewhat immature. The third future goal is to improve the learning algorithm of the model, primarily by including mutation mechanisms and the fitness function.

Acknowledgements

This paper is a revised and major expanded version of a paper entitled “Words Are Analogous To Lymphocytes: A Multi-Word-Agent Autonomous Learning Model” presented at the Twenty-Third International Conference on Systems Engineering Series: Advances in Intelligent Systems and Computing in Las Vegas of USA and during August 19-21, 2014.

References

[1] F. de Saussure, Course in General Linguistics. New York: Philosophical Library, 1959.

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

[3] N. Xue, F. Xia, F.-D. CHiou, and M. Palmer, “The Penn Chinese TreeBank: Phrase structure annotation of a large corpus,” Nat. Lang. Eng., vol. 11, no. 2, pp. 207–238, Jun. 2005.

[4] 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.

[5] M. A. K. Halliday, “Towards a language-based theory of learning,” Linguist. Educ., vol. 5, no. 2, pp. 93–116, Jan. 1993.

[6] I. R. Cohen, “Real and artificial immune systems: computing the state of the body,” Nat. Rev. Immunol., vol. 07, no. 07, pp. 569–574, Jul. 2007.

[7] J. D. Farmer, N. H. Packard, and A. S. Perelson, “The immune system, adaptation, and machine learning,” Phys. D Nonlinear Phenom., vol. 22, no. 1–3, pp. 187–204, 1986.

[8] S. Forrest, A. S. Perelson, L. Allen, and R. Cherukuri, “Self-nonself discrimination in a computer,” Proc. 1994 IEEE Comput. Soc. Symp. Res. Secur. Priv., pp. 202–212, 1994.

[9] S. F. M. Burnet, The Clonal Selection Theory of Acquired Immunity, vol. 105, no. 6. Nashville: Vanderbilt University Press, 1959.

[10] N. K. Jerne, “Towards a network theory of the immune system.,” Ann. Immunol., vol. 125C, no. 1–2, pp. 373–89, Jan. 1974.

[11] J. Greensmith, U. Aickelin, and S. Cayzer, “Introducing Dendritic Cells as a Novel Immune-Inspired Algorithm for Anomaly Detection,” in Artificial Immune Systems SE - 12, vol. 3627, C. Jacob, M. Pilat, P. Bentley, and J. Timmis, Eds. Springer Berlin Heidelberg, 2005, pp. 153–167.

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

[13] S. Wolfram, “Cellural Automaton as Models of Complexity,” Nature, vol. 311, pp. 419–424, 1984.

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

[15] I. R. Cohen, Tending Adam’s Garden: Evolving the Cognitive Immune Self, vol. 293, no. 5528. Academic Press, 2004, p. 296.

[16] M. Alber, M. Kiskowski, J. Glazier, and Y. Jiang, “On Cellular Automaton Approaches to Modeling Biological Cells,” in Mathematical Systems Theory in Biology, Communications, Computation, and Finance, vol. 134, J. Rosenthal and D. Gilliam, Eds. Springer New York, 2003, pp. 1–39.

[17] J. Liu, Autonomous Agents and Multi-Agent Systems. World Scientific Publishing Co. Pte. Ltd., 2001.

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

[19] 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.

[20] 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.

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

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

[23] X. Dong, X. Lv, Y. Guan, and J. Yang, “Multi-word-Agent Autonomy Learning Based on Adaptive Immune Theories,” Int. J. Digit. Content Technol. its Appl., vol. 7, no. 3, pp. 723–745, Feb. 2013.

[24] J. Yang, Y. Guan, and X. Dong, “Words Are Analogous To Lymphocytes: A Multi-Word-Agent Autonomous Learning Model,” in Proceedings of the Twenty-Third International Conference on Systems Engineering, 2014, pp. 755–760.

[25] J. Yang, Y. Guan, and X. Dong, “Lymphocyte-style word representations,” in 2014 IEEE International Conference on Information and Automation (ICIA), 2014, pp. 920–925.

[26] D. Dasgupta, S. Yu, and F. Nino, “Recent Advances in Artificial Immune Systems: Models and Applications,” Appl. Soft Comput., vol. 11, no. 2, pp. 1574–1587, Mar. 2011.

[27] E. Hart and J. Timmis, “Application areas of AIS: The past, the present and the future,” Appl. Soft Comput., vol. 8, no. 1, pp. 191–201, Jan. 2008.

[28] D. Dasgupta, “An Overview of Artificial Immune Systems and Their Applications,” in Artificial Immune Systems and Their Applications, D. Dasgupta, Ed. Springer Berlin Heidelberg, 1999, pp. 3–21.

[29] L. N. de Castro and F. J. Von Zuben, “Learning and optimization using the clonal selection principle,” IEEE Trans. Evol. Comput., vol. 6, no. 3, pp. 239–251, Jun. 2002.

[30] A. Watkins, J. Timmis, and L. Boggess, “Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm,” Genet. Program. Evolvable Mach., vol. 5, no. 3, pp. 291–317, Sep. 2004.

[31] G. Dudek, “An Artificial Immune System for Classification With Local Feature Selection,” IEEE Trans. Evol. Comput., vol. 16, no. 6, pp. 847–860, Dec. 2012.

[32] L. N. De Castro and F. J. Von Zuben, “aiNet: An Artificial Immune Network for Data Analysis,” in Data Mining A Heuristic Approach, H. A. Abbass, R. A. Sarker, and C. S. Newton, Eds. Idea Group Publishing, 2001, pp. 231–259.

[33] J. Hunt, “Learning using an artificial immune system,” J. Netw. Comput. Appl., vol. 19, no. 2, pp. 189–212, 1996.

[34] Z. Deng, G. Tan, P. He, and J. Ye, “A decision hyper plane heuristic based artificial immune network classification algorithm,” J. Cent. South Univ., vol. 20, no. 7, pp. 1852–1860, Jul. 2013.

[35] Y. Zhong and L. Zhang, “An Adaptive Artificial Immune Network for Supervised Classification of Multi-/Hyperspectral Remote Sensing Imagery,” IEEE Trans. Geosci. Remote Sens., vol. 50, no. 3, pp. 894–909, Mar. 2012.

[36] M. F. Abbod and S. Alsharhan, “Artificial Immune Systems – Models, Algorithms and Applications,” Comput. Eng., vol. 3, no. May, pp. 118–131, 2010.

[37] G. Figueredo, U. Aickelin, and P.-O. Siebers, “Systems Dynamics or Agent-Based Modelling for Immune Simulation?,” in Artificial Immune Systems SE - 10, vol. 6825, Springer Berlin Heidelberg, 2011, pp. 81–94.

[38] N. Fachada, V. Lopes, and A. Rosa, “Agent-based modelling and simulation of the immune system: a review,” EPIA 2007 LNCS, vol. 4874, pp. 300–315, 2007.

[39] F. Celada and P. E. Seiden, “A computer model of cellular interactions in the immune system.,” Immunol. Today, vol. 13, no. 2, pp. 56–62, Feb. 1992.

[40] M. Bernaschi and F. Castiglione, “Design and implementation of an immune system simulator.,” Comput. Biol. Med., vol. 31, no. 5, pp. 303–331, Sep. 2001.

[41] G. Figueredo, P.-O. Siebers, U. Aickelin, and S. Foan, “A Beginner’s Guide to Systems Simulation in Immunology,” in Artificial Immune Systems SE - 5, vol. 7597, C. Coello Coello, J. Greensmith, N. Krasnogor, P. Liò, G. Nicosia, and M. Pavone, Eds. Springer Berlin Heidelberg, 2012, pp. 57–71.

[42] W. R. Ashby, “Principles of the self-organizing system,” in in Principles of Self-Organization: Transactions of the University of Illinois Symposium, 1962, pp. 255–278.

[43] J. Liu, S. Zhang, and J. Yang, “Characterizing Web Usage Regularities with Information Foraging Agents,” IEEE Trans. Knowl. Data Eng., vol. 16, no. 5, pp. 566–584, 2004.

[44] S. Zhang and J. Liu, “From Local Behaviors to the Dynamics in an Agent Network,” in Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence, 2006, pp. 572–580.

[45] K. C. Tsui and J. Liu, “Multiagent Diffusion and Distributed Optimization,” in Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems, 2003, pp. 169–176.

[46] X.-F. Xie and J. Liu, “How Autonomy Oriented Computing (AOC) Tackles a Computationally Hard Optimization Problem,” in Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems, 2006, pp. 646–653.

[47] B. Yang, J. Liu, and D. Liu, “An Autonomy-oriented Computing Approach to Community Mining in Distributed and Dynamic Networks,” Auton. Agent. Multi. Agent. Syst., vol. 20, no. 2, pp. 123–157, 2010.

[48] B. Yang and J. Liu, “Discovering Global Network Communities Based on Local Centralities,” ACM Trans. Web, vol. 2, no. 1, pp. 9:1–9:32, 2008.

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

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

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

[52] J. Nivre, J. Hall, and J. Nilsson, “MaltParser: A data-driven parser-generator for dependency parsing,” in Proceedings of LREC, 2006, vol. 6, pp. 2216–2219.

[53] Y. Zhang and S. Clark, “A tale of two parsers,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing - EMNLP ’08, 2008, p. 562.

[54] R. McDonald and J. Nivre, “Analyzing and integrating dependency parsers,” Comput. Linguist., vol. 37, no. 1, pp. 197–230, 2011.

[55] R. McDonald, K. Crammer, and F. Pereira, “Online large-margin training of dependency parsers,” in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL ’05, 2005, pp. 91–98.

[56] J. Yang, Y. Guan, X. Dong, and B. He, “Representing Words as Lymphocytes,” in Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 3146–3147.

[57] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in Advances in Neural Information Processing Systems, C. j. c. B. and L. B. and M. W. and Z. G. and K. q. Weinberger, Ed. 2013, pp. 3111–3119.