International Journal of Multimedia and Ubiquitous Engineering
Vol.10, No.4 (2015), pp.355-366
http://dx.doi.org/10.14257/ijmue.2015.10.4.34

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

Jinfeng Yang[1], Yi Guan[*1] and 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 language network is then simulated as an immune network. Meanwhile, Spreading Activation 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. In these interactions, some words depend on or are depended upon by others. In the dependency tree of a sentence, 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. Everything in a language can be formally described in terms of nodes and edges between nodes [1], 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

ISSN: 1975-0080 IJMUE

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 [2]. With this understanding, regulating or learning the combination strength between words is of key importance in sentence structure analysis.

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 [2]. Second, spreading activation[3] 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 [1]. 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) [4] in the autonomy oriented computing (AOC) [5] framework is employed to construct the model. The immune system is composed of autonomous lymphocytes, *i.e.*, a type of immune system cell [6]. As an agent-based modeling method, CA is a natural application for modeling cellular systems [7]. Being a generic framework, AOC offers a new computing paradigm that makes use of autonomous entities for solving computational problems and modeling complex systems[8]. 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 the Penn Chinese Treebank 5.1 (CTB) [9] indicate that this model can effectively and continuously learn word combination strength.

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

2. Related Work

Analogies between words and lymphocytes were first proposed by Dong [10]. Inspired by Dong’s work, Yang [11] developed a revised model and proposed a lymphocyte-style word representation [12]. However, both the two models did not employ the immune network theory comprehensively. As a major extension of Yang [11], 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 about immune-based learning and agent-based modeling.

2.1. Immune-based Learning

To date, there have been four types of AIS algorithms used in applied AIS[13]: 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 [14]. Clonal selection theory has inspired the unsupervised learning model CLONALG [15] and the supervised AIS classifier AIRS [16]. Idiotypic immune network theory was first proposed by Jerne [17] and formalized into a model by Farmer [18]. 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 [17, 19]. By employing the metaphor of immune network theory, unsupervised learning models were also proposed in[20]. After Hunt and Cooke first attempted to introduce immune network algorithms into a supervised binary classifier [21], many AIS classifiers were proposed [22, 23].

2.2. Agent-based Modeling

Agent-based modeling 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). One of the most referenced IS simulators, ImmSim, was based on CA with probabilistic rules [24]. 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 [25].

Autonomy Oriented Computing (AOC) is a generic and formulated framework for modeling multi-agent systems [5]. 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 [26]. The AIS model of this research follows the idea of C-ImmSim and is developed under the AOC framework.

3. Multi-word-agent Autonomous Learning Model

3.1. 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.

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.

The most key import elements of the MWAALM are representations, evaluation function and behavior of word agents, and the system objective function. The formulated model is described in detail in the Appendix[1].

3.2. Representations of Word Agents

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.

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 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.

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).

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

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

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 (3).

$$E^w = \{df_1^w, df_2^w, \cdots, df_i^w, \cdots df_{N_{df}^w}^w\} \qquad (3)$$

3.3. 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 (4)

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

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

[1] https://github.com/yangjinfeng/wordrep/blob/master/mwaalm_appendix.pdf

3.4. Behaviors

A BWA has five primitive behaviors: moving, recognition, spreading-activation, cloning and hypermutation. A BWA can move randomly to adjacent sites or stay where it resides. When the 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 $L_{activation}$. The stimulated BWA w can act like an antigen and transfer its activation value to another BWA, 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. 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. 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):

$$\begin{gathered}\omega_i'^{w'} = \omega_i^{w'} + \Delta_i^{w'}, \\ \Delta_i^{w'} = \alpha * (1/\beta) * N(0,1), \\ \alpha = \exp(-f_{affinity}) * \exp(-f_{fitness}(w')) \quad (6)\end{gathered}$$

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), 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.

3.5. System Objective Function

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 [2]. 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.

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 (7) and (8). The system objective function Φ combines f_{UAS} and f_{score} and is defined using Equation (10).

$$f_{UAS}(T',T)=\frac{|E'\cap E|}{|E|} \quad (7)$$

$$f_{score}(T',T)=\frac{score(T)}{score(T')} \quad (8)$$

$$score(T)=\sum_{e\in E} score(e)=\sum_{e\in E} f_{affinity}(w_{head}^e, w_{dependent}^e) \quad (9)$$

$$\Phi(w_1^S, w_2^S, \cdots, w_i^S, \cdots, w_{N_S}^S)=f_{UAS}(T',T) * f_{score}(T',T)=\frac{|E'\cap E|}{|E|} * \frac{score(T)}{score(T')} \quad (10),$$

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 (11).

$$f_{fitness}(w')=\Phi(w_1^S, w_2^S, \cdots, w', \cdots, w_{N_S}^S) \quad (11)$$

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.

$$w'^* = \arg\max_i (f_{fitness}(w_i')) \quad (12)$$

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. Experimental Results

4.1. 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 [27] and the head-finding rules [28] 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. The performance of the dependency parsing of the model is evaluated by the UAS on the test set, which is defined as equation (13).

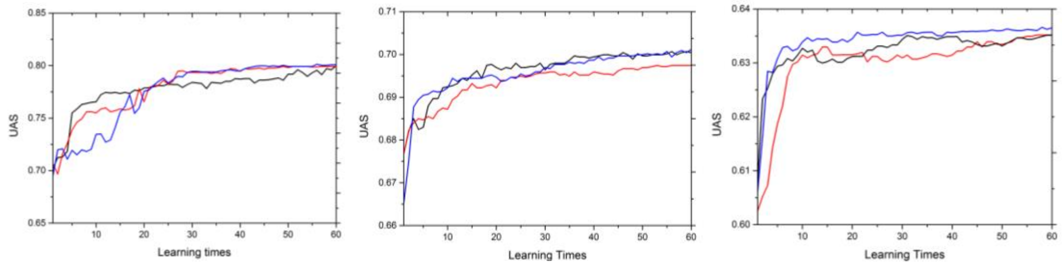
$$UAS = \frac{\#of\ \ words\, that\, are\, assigned\, correct\, heads\, in\, the\, test\, set}{\#of\ \ words\, in\, the\, test\, set} \qquad (13)$$

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[29]. 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 [30], which is a graph-based parser that also uses the MST algorithm [2]. 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.

4.2. 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.

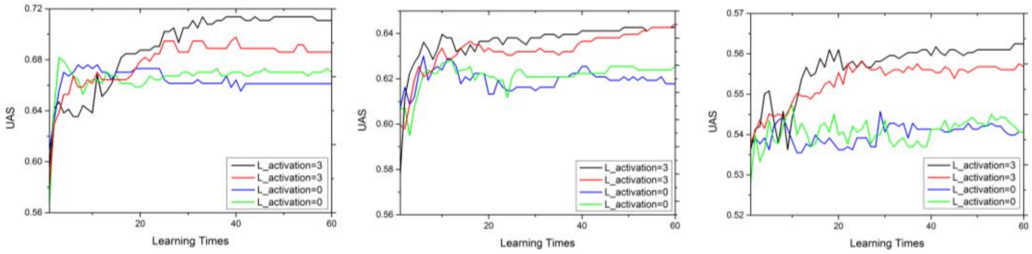
4.2.1 Performance of the MWAALM for Dependency Parsing: The experimental results on three larger-scale data sets with different sentence lengths are shown in Figure 1. In these experiments, the parameters of the model are set to the same values, and $L_{activation}$ is set to 3. Although the three groups of result curves in Figure 1 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 1. Results on Three Larger-scale Data Sets

4.2.2 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 Figure 2. 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.

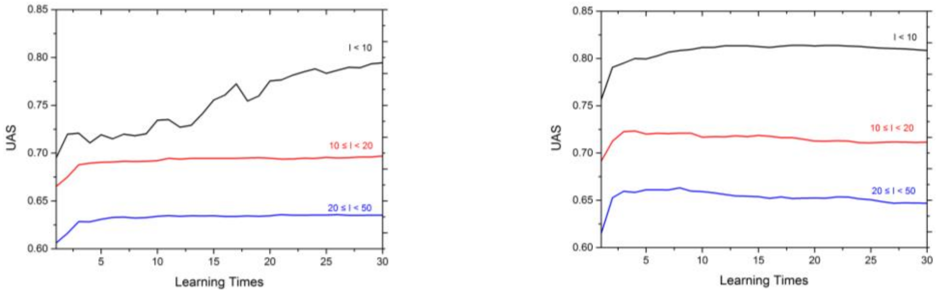


(a) Results on the short-sentence ($l < 10$) datat set (b) Results on the medium-length-sentence ($10 \le l < 20$) datat set (c) Results on the long-sentence ($20 \le l < 50$) datat set

Figure 2. Comparative Results with Different Activation Levels on Three Smaller-scale Data Sets

4.2.3. 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.

The two models are compared with different sentence-length data sets, as shown in Figure 3. 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.



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

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

Figure 3. Comparisons of Performances between MWAALM and MSTParser on Different Sentence-length Data Sets

The final results generated by MWAALM and MSTParser on three larger data sets are displayed in Table 1. 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 1. 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

5. 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 [31] and has the potential to express combinative relations, which is an inherent limitation of existing word representations such as distributed word representation [32]. 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.

References

[1] R. Hudson, “An Introduction to Word Grammar”. Cambridge University Press, **(2010)**.

[2] J. M. Eisner, “Three new probabilistic models for dependency parsing: An exploration,” Proceedings of the 16th Conf. Comput. Linguist, **(1996)**, vol. 1, pp. 340–345.

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

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

[5] K. C. T. Jiming Liu and X. Jin, “Autonomy Oriented Computing: From Problem Solving to Complex Systems Modeling”, Springer, **(2005)**.

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

[7] 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.

[8] J. Liu, “Autonomous Agents and Multi-Agent Systems”. World Scientific Publishing Co. Pte. Ltd., **(2001)**.

[9] N. Xue, F. Xia, F.-D. CHiou, and M. Palmer, “The Penn Chinese TreeBank: Phrase structure annotation of a large corpus,” Nat. Lang. Eng., **(2005)**, vol. 11, no. 2, pp. 207–23.

[10] 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., **(2013)**, vol. 7, no. 3, pp. 723–745.

[11] 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.

[12] 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.

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

[14] S. F. M. Burnet, “The Clonal Selection Theory of Acquired Immunity”, Nashville: Vanderbilt University Press, vol. 105, no. 6, **(1959)**.

[15] 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, **(2002)**, pp. 239–251.

[16] 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, v pp. 291–317.

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

[18] 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, **(1986)**, pp. 187–204.

[19] A. S. Perelson, “Immune network theory,” Immunol. Rev., vol. 110, **(1989)**, pp. 5–36.

[20] 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.

[21] J. Hunt, "Learning using an artificial immune system," J. Netw. Comput. Appl., vol. 19, no. 2, **(1996)**, pp. 189–212.

[22] 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, **(2013)**, pp. 1852–1860.

[23] 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, **(2012)**, pp. 894–909.

[24] F. Celada and P. E. Seiden, "A computer model of cellular interactions in the immune system.," Immunol. Today, vol. 13, no. 2, **(1992)**, pp. 56–62.

[25] M. Bernaschi and F. Castiglione, "Design and implementation of an immune system simulator.," Comput. Biol. Med., vol. 31, no. 5, **(2001)**, pp. 303–331.

[26] 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.

[27] [27] J. Nivre, J. Hall, and J. Nilsson, "MaltParser: A data-driven parser-generator for dependency parsing," in Proceedings of LREC, vol. 6, **(2006)**, pp. 2216–2219.

[28] 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.

[29] [29] R. McDonald and J. Nivre, "Analyzing and integrating dependency parsers," Comput. Linguist., vol. 37, no. 1, **(2011)**, pp. 197–230.

[30] 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.

[31] 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.

[32] 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.

Author

Jinfeng Yang, he received the B.S. degree in mechanical engineering from Xian Shiyou University in 2002 and the M.S. degree in computer application technology from Xian Shiyou University in 2006. Now he is a Ph.D. candidate at School of computer science and technology in Harbin Institute of Technology, and his research interests cover natural language processing, machine learning and information extraction on electronic medical records.