Representing Words as Lymphocytes: Appendix

Jinfeng Yang, Yi Guan, Xishuang Dong, Bin He
yangjinfeng2010@gmail.com, guanyi@hit.edu.cn,{dongxishuang, goohebingle}@gmail.com
School of Computer Science and Technology of Harbin Institute of Technology
Harbin, Heilongjiang, China 150001

A0 Summary of the Appendix

This appendix serves as a supplement to the poster submission, mainly involves three aspects: inspirations from immune system, learning of the word representations and experimental results. In this research, a lymphocyte-style word representation is proposed inspired by the analogies between words and lymphocytes. For learning of the representations, a multi-word-agent autonomous learning model (MWAALM) based on an artificial immune system is presented. The model is constructed by Cellular Automation, and words are modeled as B cell word agents (BWAs) and as antigen word agents (AgWAs).

A1 Inspirations from the Immune System

The adaptive immune system (IS) serves to protect the human body against foreign pathogens. When pathogens, also called antigens (Ags), invade the human body, bone cells (B cells), an important class of lymphocytes in the IS, recognize Ags by their receptors and then undergo a sequence of state changes resulting in higher affinities between the B cells and Ags. The adaptive immune system is composed of plenty of 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. In each site of the grid, the simulative lymphocytes reside 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. The higher concentrations of B cells mean the higher the importance of the B cells, since they need to recognize more Ags. B cells can recognize specific Ags with their receptors and then can be stimulated by the Ags. The receptors of B cells have a Y-shaped structure (Kuby, Kindt, and et al. 2002), 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. So, the paratopes of one B cell can also recognize the idiotopes of another B cell. The two variable regions at the tips of Y are identical. For simplicity in our model, the left tips of the Y are idiotopes and the right tips are paratopes. 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' (de Saussure 1959). During interacting, some words depend on others or are depended upon by others. Moreover these dependency relations can be represented as a word network, also called a language network (Hudson 2010). 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 excerpted from Penn Chinese Treebank 5.1(CTB) (Xue, Xia, and et al. 2005), 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 on which it depends, called the head. Dependency relations are also called head-dependent pairs represented by arrows pointing from the head to the dependent. Each word in a sentence both resides in a dominant context and a dependent context. So word properties may need to be grouped into head properties and dependent properties. Figure 2 shows the simplified design of B cell receptors. In this research, B cells represent words and are modeled as BWAs, B cells concentrations represent words frequencies, and B cell receptors represent word properties (or features). Dependency features extracted from annotated head-dependent pairs

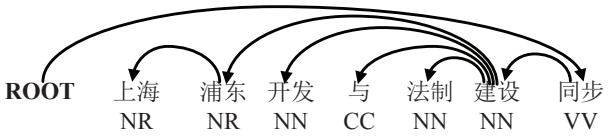


Figure 1: The dependency structure for a Chinese sentence.

are used as word properties and are grouped into dependent properties, corresponding to idiotops, and head properties, corresponding to paratopes.

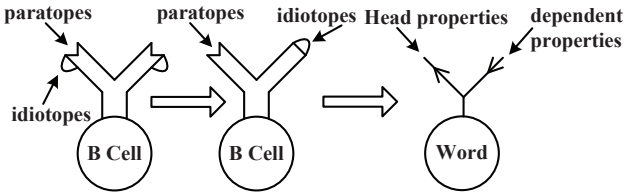


Figure 2: The simplified design of B cell receptors

In the immune response, the only role of Ags is to match and stimulate B cells and get killed. Ags match B cells' paratopes with their unique set of antigenic determinants also called epitopes. In the proposed model, Ags are modeled as AgWAs, and epitopes are 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 strengths are calculated based on the similarity between head properties and dependent properties, accumulating weights of the matched properties. The initial weights of word properties can be set to zero or 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 (Hart, Bersini, and Santos 2009); and according to simulation, a power-law degree-distribution can emerge (Hart 2006). Similar to an immune network regarding its complexity, the language network is also a complex network (Hudson 2010). As for the similarity between idiotypic immune network and 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 whole network and may stop at certain criterion, such as the maximum idiotypic level (Perelson 1989). Spreading activation is used in cognitive psychology to model the fan out effect (Collins and Loftus 1975) and is also used as the memory mechanism of humans using a language network (Hudson 2010). In this AIS model, spreading activation is employed as the mechanism of idiotypic interaction 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 strengths. When an AgWA stimulates a BWA (e.g. B cell 1), the BWA is activated and is assigned an initial active value (e.g. 3). This initial active value is called activation level, which determines the spreading depth and can spread along the inverse direction of arcs. The activated BWA can stimulate other dependent BWA(e.g. B cell 3), which has the most 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.

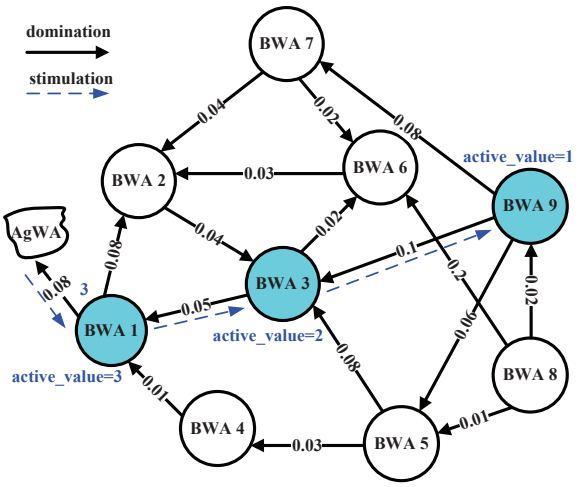


Figure 3: The process of spreading activation in the artificial immune network of this model.

During the period of immune response, B cells concentration may change because B cells stimulated by Ags can reproduce themselves. In our model, we aim to regulate words strengths, so for simplicity B cells concentrations will not really change and clonal expansion is applied to generate a number of offspring used as candidates from which, the best one is to be selected. The number of clonal candidates is a parameter of our model. Hypermutation is the most

important mechanism which is suffered by the offspring and promise to generate more powerful B cells. In our model, hypermutation is introduced at the matched properties. 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, the smaller the mutation rate (De Castro and Timmis 2002)).

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

Based on these inspirations, Agents, including BWAs and AgWAs, and the environment of MWAALM are designed. The components of adaptive immune theory and their counterparts in MWAALM are summarized in Table 2 below.

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

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

A2 Learning of Word Representations

Outline

In order to train and learn the lymphocyte-style word representations, a Chinese dependency Treebank is used as training data. The learning of the model is equivalent to the artificial immune response. The process of learning or immune response of this AIS model includes two 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. BWAs are built from the training set, as well as their receptors and the artificial immune network; then BWAs are distributed into the grid uniformly. In the learning stage, AgWAs are constructed from a sentence from the dependency Treebank one by one and are injected into the immune environment. With the principles of clonal selection and idiotypic immune network, BWAs and AgWAs interact with each other resulting in mutated properties' weights. The learning process of the model simulates the process of immune response and also is accordance with the framework of online learning (McDonald, Crammer, and Pereira 2005).

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 according to their affinities.
3.3: BWAs make clones.
3.4: Clones undergo hypermutation.
3.5: Mutated clones are evaluated by a fitness function and the best fit one is reserved.
3.6: The reserved BWAs act as antigens in the artificial immune network.

The pseudo code of the MWAALM model is given in Algorithm 1.

Environment

The environment of MWAALM is an $M \times M$ grid, in which each site is surrounded by eight adjacent sites. More than one word agent can reside in a site and can move to one of the adjacent sites freely and randomly. Figure 4 shows the design of the environment.

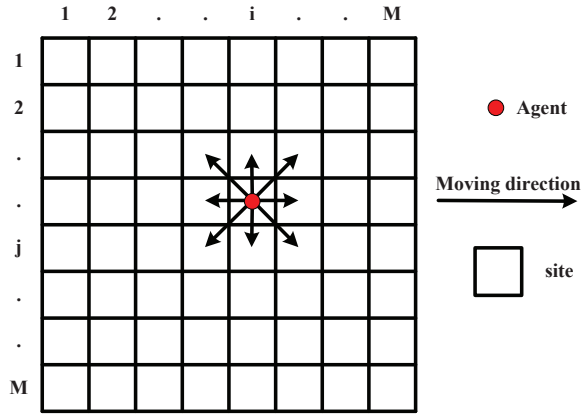


Figure 4: The design of the environment.

The environment is characterized by an attribute set $\varepsilon s = \{es_1, es_2, \cdots, es_{N_{\varepsilon s}}\}$, where each attribute corresponds to a unique word property, and $N_{\varepsilon s}$ denotes the number of all unique word properties. So εs is a shared 'notice board,' by which word agents post their new property values or read new property values posted by other word agents. 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.

Representation of BWA

Words in training set are represented as B cells and modeled as BWAs, i.e. lymphocyte-style Word Representations. Head properties and dependent properties of words are represented as paratopes P^w and idiotopes I^w on receptors of BWA w respectively, formulated as Equation (1) and Equation (2) in the poster paper. Dependency features extracted from head-dependent pairs are used as properties of words according to the feature templates shown as Table 2.

Table 2: Feature templates of dependency pairs

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 donates the head word, W_d donates the dependent word, P_h donates the POS of the head word, P_d donates the POS of the dependent word, $+1$ donates the right adjacent word, -1 donates the left adjacent word. For example, 法制(legal)←建设(construction) is a head-dependent pair in the dependency tree shown in Figure 1, and NN and JJ are their corresponding POS tagged below them. Then features of the head-dependent pair include *法制, 建设, 法制_建设, JJ, NN, JJ_NN*, etc.

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 epitopes of the AgWA. The epitopes E^w of an AgWA w is formulated as Equation (1).

$$E^w = \{df_1^w, df_2^w, \cdots, df_{N_E}^w\} \quad (1)$$

Affinity Measurement B cells recognize Ags or other B cells according to affinities between them. Affinities between BWAs are also the strengths of the dependency relations between them. Affinity or strength is calculated based on the similarity between paratopes and epitopes or idiotopes, accumulating weights of the matched properties. Affinity between a BWA w_B and an AgWA w_{Ag} is measured by equation (2), and affinity between a BWA w_B and

other BWA $w_{B'}$ is measured by equation (3).

$$f_{aff}(w_B, w_{Ag}) = \sum_{i=1}^{N_P^{w_B}} \sum_{j=1}^{N_E^{w_{Ag}}} \delta\left(hf_i^{w_B}, df_j^{w_{Ag}}\right) \omega_{hf_i^{w_B}} \tag{2}$$

$$f_{aff}(w_B, w_{B'}) = \sum_{i=1}^{N_P^{w_B}} \sum_{j=1}^{N_I^{w_{B'}}} \delta\left(hf_i^{w_B}, df_j^{w_{B'}}\right) \omega_{hf_i^{w_B}} \tag{3}$$

$$\delta(x, y) = \begin{cases} 1, & \text{if } x = y \\ 0, & \text{otherwise} \end{cases} \tag{4}$$

Provided with well-tuned strengths of the dependency relations between two words in a sentence, the sentence's dependency structure can be created in a bottom-up paradigm (Eisner 1996). Therefore, in the learning stage, the regulations of the weights of words' properties are evaluated according to the unlabeled attachment score (UAS) (Kubler, McDonald, and Nivre 2009) of the predicted dependency tree of the training sentence, i.e. the percentage of words that have the correct heads.

Behaviors of Word Agents

Behaviors of a BWA include moving, recognition, spreading activation, cloning, and hypermutation. Moving is the only behavior of an AgWA. By their behaviors, word agents can only interact with their neighbor agents. In this model, the neighbors of a word agent w are a group of word agents $L^w = \{l_1^w, l_2^w, \cdots, l_i^w, \cdots, l_{N_L}^w\}$, where l_i^w resides in the same site of the grid as w, and $l_{N_L}^w$ is the number of neighbors.

Moving: Both BWAs and AgWAs have the same moving behaviors; they can move randomly to adjacent sites or stay where they reside.

Recognition: A BWA recognizes other neighbor AgWA with its paratopes according to affinity between them, then the BWA is activated and is assigned an initial integer active value as activation level $L_{activation}$.

Spreading activation: The activated BWA can act like an antigen, and transfer its active value to another BWA with the active value weakened by one. The process of activation propagation continues until the active value of the B cell word agent descends 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 just comes down to a clonal selection algorithm.

Cloning: Once a BWA w is activated 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. The process of hypermutation is that the weightiness $\omega_i^{w'}$ 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 weight and fitness of the agent and also inversely proportional to the affinity between the agent and the recognized AgWA. The mutation is performed according to equation (5):

$$\begin{aligned} {\omega'}_i^{w'} &= \omega_i^{w'} + \Delta_i^{w'}, \\ \Delta_i^{w'} &= \alpha \times (1/\beta) \times N(0,1), \\ \alpha &= e^{-\omega_i^{w'}} \times e^{-f_{aff}} \times e^{-f_{fitness}(w')} \end{aligned} \tag{5}$$

where ${\omega'}_i^{w'}$ is the mutated weightiness, $N(0,1)$ 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_{aff} is the affinity determined by the equation (4) or (5), and $f_{fitness}(w')$ is the fitness of each clone determined by a fitness function. Since $\Delta_i^{w'}$ is always greater than zero, ${\omega'}_i^{w'}$ is always greater than $\omega_i^{w'}$. According to equation (4) or (5), the affinity will be higher than before and the agent will reach its goal, but not all clones will be reserved. These clones will be evaluated by a fitness function and then the best fit one will be reserved and replace its parent. In the model, two initialization modes are considered to initialize the value of ω_i^w. The first mode is to initialize the value of ω_i^w as $(1/\beta) \times N(0,1)$, and the second is to initialize the value of ω_i^w as zero.

Fitness Function

When a clone w' of the BWA w finishes its hypermutation, the weightiness of the paratopes of its receptor may be changed and word strength between w' and the AgWA may be regulated. The fitness of w' determines whether the word combination strengths are tuned better or worse.

The fitness function is designed as a UAS function, formulated as equation (6), for the predicted dependency tree of the training sentence from which antigens are built. The dependency tree prediction is implemented by using the MST algorithm[25]. Let T be the training sentence, be the annotated dependency tree, and be the predicted dependency tree,

then the fitness of w' is measured by the equation (7).

$$UAS = \frac{\#words\ with\ correct\ assigned\ heads}{\#words\ in\ training\ set} \quad (6)$$

$$f_{fitness}(w') = f_{UAS}(T, T') \quad (7)$$

According to equation (8), 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 replaces its parent and others are eliminated.

$$w'^* = \arg\max_i(f_{fitness}(w'_i)) \quad (8)$$

If $f_{fitness}(w'^*) > f_{fitness}(w)$ then w is replaced by w'^*, otherwise w is still replaced by w'^* 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 towards the desired state in which combination strengths between words are well tuned.

The learning process of the MWAALM is visualized as shown in figure 5. In figure 5, blue dots denote BWAs, green dots denote active BWAs, and red dots denote AgWAs.

A3 EXPERIMENTAL RESULTS

Data sets and Experimental Design

Lymphocyte-style word representations can be learned on a dependency parsed data set and a dependency Treebank , built from Penn Chinese Treebank 5.1(CTB) by applying Penn2Malt tool (Nivre, Hall, and Nilsson 2006), is employed as experimental data. All words of sentences in the converted dependency Treebank were used to initialize B cells and dependency relations between words were used to initialize the artificial immune network. Idiotopes and paratopes of B cells were equipped by features of head-dependent pairs. The proposed word representations are evaluated by computing word similarities.

Results of Word similarity Computing

For evaluation of the proposed word representations, words in the first 100 sentences of the CTB are considered. The total number of these words is 838. According to equation (11), similarity between each considered word and each word in the Treebank are computed. For each considered word, five words with most high similarities are chosen for evaluation.

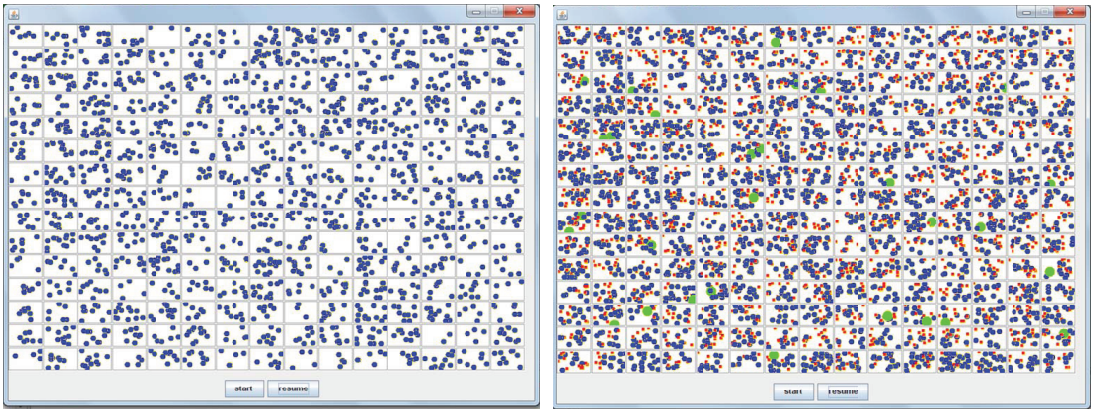
Exemplars of learned word representations Three examples of word representations learned from the CTB are shown in Figure 6. Each word is represented as two vectors, corresponding to a head property vector and a dependent property vector. Each dimension corresponds to a specific property and its value corresponds to the weight of the property. In Figure 6, 法制(legal) depends on 建设(construction), and this dependency relation is also shown in Figure 1. The strength of dependency relation between 法制(legal) and 建设(construction) can be computed between the head properties of 法制(legal) and the dependent properties of 法制(legal) according to equation (3). The 调整(adjustment) is similar to 建设(construction), and the similarity between 建设(construction) and 调整(adjustment) can be computed by adopting cosine similarity on both head properties and dependent properties according to Equation (3) in the poster paper. Table 3 lists several considered words as exemplars with different POS and five candidate similar words. According to the judgment used in this research, these candidate similar words with most high similarities seem to be believable.

Evaluated results for word similarity Two precision metrics are used to evaluate those mined similar words. The one is the precision of top one P_{Top1}, which means the percentage of those considered words whose top one candidate word is judged similar. The second is the precision of top five P_{Top5}, which means the percentage of those considered words for which one of the top five candidate words is judged similar. For the purpose of impartial evaluation, two persons evaluated the mined candidate similar words of the 838 words independently.

In this research, word representations are initialized randomly and learned or optimized by the MWAALM. Each property's weight ω_i^w of a word w is initialized as $(1/\beta) \times N(0, 1)$, To validate the effectiveness of the MWAALM, word similarity evaluations is also conducted on the initialized word representations.

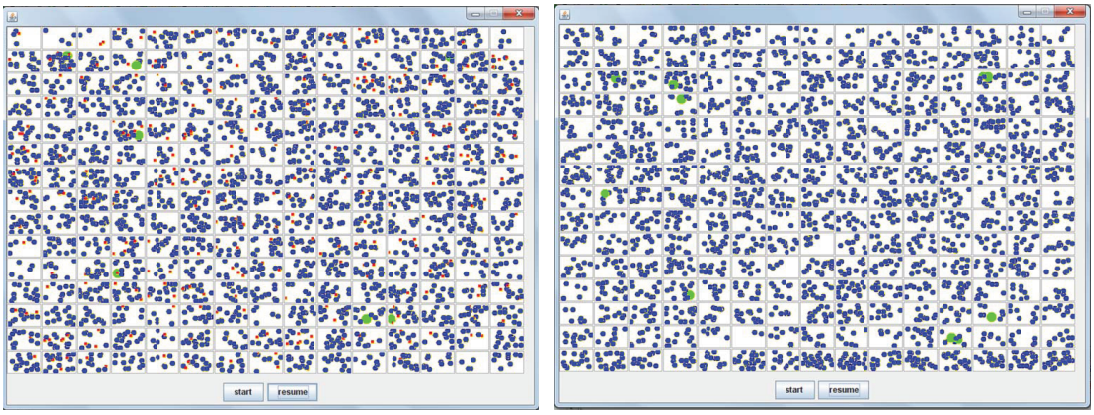
The two evaluated files for initialized word representations are init_sim_1.xls and init_sim_2.xls, and the two evaluated files for learned word representations are learned_sim_1.xls and learned_sim_2.xls. These evaluated files can be downloaded from the page[1] of my Github. According to evaluation, the values of the two precision metrics evaluated by two persons on both initialized and learned

[1] https://github.com/yangjinfeng/wordrep/evaluation



(a) The immune environment is initialized

(b) AgWAs are injected into the immune environment and some BWAs are activated



(c) Some AgWAs are killed

(d) All AgWAs are cleaned from the immune environment

Figure 5: Visualizations of the learning process of the MWAALM

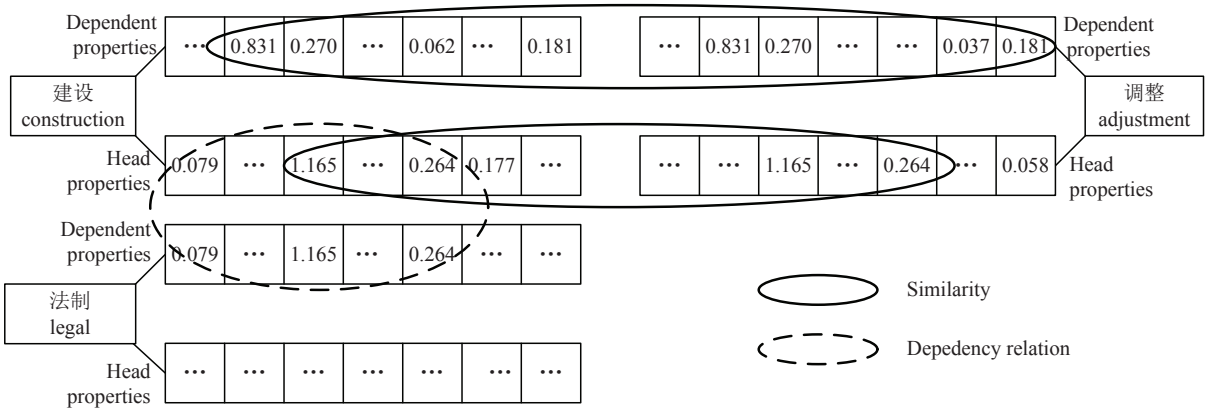


Figure 6: Examples of word representations learned from the CTB.

Table 3: Examples of considered words and their five candidate similar words with highest similarities.

Considered word	**Similar word 1**	**Similar word 2**	**Similar word 3**	**Similar word 4**	**Similar word 5**
十月_NT October	十一月份(0.9416) November	十一月(0.7227) November	八月(0.6922) August	年初(0.6895) earlier year	上午(0.6713) morning
企业_NN enterprise	项目(0.7729) project	开发区(0.7646) developing	政策(0.7571) policy	工程(0.7570) project	公司(0.7444) enterprise
比较_AD more or less	古来(0.9798) ever since long ago	较为(0.9793) more or less	日臻(0.9769) more and more	最为(0.972) most	相对(0.972) relative
中国_NR China	上海(0.5774) Shanghai	美国(0.5397) America	天津(0.5131) Tianjin	北京(0.509) Beijing	东南亚(0.503) East south Asia
中_LC middle	后(0.7321) back	间(0.7022) middle	内(0.6574) inner	上(0.647) upper	外(0.5457) outer
家_M one (hotel, shop, etc.)	元(0.685) yuan(￥)	美元(0.6775) dollar($)	吨(0.6743) ton	个(0.6121) one (apple, egg, etc.)	项(0.5803) item

word representations are shown in table 4. The evaluation results by two persons seem to be in high agreement. As shown in table 4, the results on learned representations have gained much high improvements than that on initialized representations. The results indicate that the proposed lymphocyte-style word representation can be successfully applied for word similarity computing and is proven to be an effective word representation, and the MWAALM presents its promise ability.

Table 4: The evaluation results of similar words.

Representations	Evaluator 1		Evaluator 2	
	P_{Top1}	P_{Top5}	P_{Top1}	P_{Top5}
Initialized	0.5120	0.6647	0.5274	0.6575
Learned	0.6337	0.7864	0.6050	0.7840

References

Collins, A., and Loftus, E. 1975. A spreading-activation theory of semantic processing. *Psychological Review* 82(6):407–428.

De Castro, L., and Timmis, J. 2002. *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer.

de Saussure, F. 1959. *Course in General Linguistics*. New York: Philosophical Library.

Eisner, J. 1996. Three new probabilistic models for dependency parsing: An exploration. *Proceedings of the 16th conference on Computational linguisticsVolume 1* 96(August):340–345.

Hart, E.; Bersini, H.; and Santos, F. 2009. Structure versus function: a topological perspective on immune networks. *Natural Computing* 9(3):603–624.

Hart, E. 2006. Analysis of a Growth Model for Idiotypic Networks. In Bersini, H., and Carneiro, J., eds., *Artificial Immune Systems SE - 6*, volume 4163 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 66–80.

Hudson, R. 2010. *An Introduction to Word Grammar*. Cambridge University Press.

Kubler, S.; McDonald, R.; and Nivre, J. 2009. Dependency Parsing. *Synthesis Lectures on Human Language Technologies* 2(1):1–127.

Kuby, J.; Kindt, T.; and et al., B. O. 2002. *Kuby Immunology*. New York: W.H. Freeman and Company, fifth edition.

McDonald, R.; Crammer, K.; and Pereira, F. 2005. Online large-margin training of dependency parsers. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics - ACL '05*, 91–98. Morristown, NJ, USA: Association for Computational Linguistics.

Nivre, J.; Hall, J.; and Nilsson, J. 2006. MaltParser: A data-driven parser-generator for dependency parsing. In *Proceedings of LREC*, volume 6, 2216–2219. Citeseer.

Perelson, A. 1989. Immune network theory. *Immunological reviews* 110:5–36.

Xue, N.; Xia, F.; and et al., F. C. 2005. The Penn Chi-

nese TreeBank: Phrase structure annotation of a large corpus. *Natural Language Engineering* 11(2):207–238.