

A Syntax Parsing Method Based on Adaptive Genetic Annealing Optimization HMM

Rong Li* and Hong-bin Wang

*Department of computer
Xinzhou Teachers' University, Xinzhou, 034000, China*

lirong_1217@126.com

Abstract

In order to further enhance the performance of syntax parsing, for the shortcomings of hidden Markov model (HMM) in the parameter optimization, an improved syntax parsing method based on adaptive genetic annealing and HMM was presented. First, an adaptive hybrid genetic annealing algorithm was adopted to optimize HMM initial parameters. Second, the improved HMM was trained by Baum Welch algorithm, and then a modified Viterbi algorithm was used to recognize various types of phrases at the same layer, finally a hierarchical analysis algorithm and Viterbi algorithm were combined together to solve hierarchy and recursion in the sentence. In the adaptive genetic annealing HMM algorithm, genetic operators and parameters of simulated annealing (SA) were first respectively improved, subpopulations were classified according to the adaptive crossover and mutation probability of GA in order to realize the multi-group parallel search and information exchange, which could avoid premature and accelerate convergence, then SA was taken as a GA operator to strengthen the local search capability. Compared with several new approaches, $F_{\beta=1}$ value is averagely increased by 3%. The experiment results prove that this method is very effective for syntactic parsing.

Keywords: *syntax parsing, genetic annealing, hidden Markov model, Viterbi algorithm*

1. Introduction

Syntactic parsing can be divided into two categories, shallow and deep parsing. Shallow parsing includes word segmentation, named entity recognition, and chunk recognition technology, *etc.* Deep parsing is based on shallow parsing, and combined with syntax, semantics and pragmatics. Syntactic analysis is a key link and difficulty in natural language understanding, and is an important preprocessing method of automatic thesaurus construction, text retrieval, text classification, information extraction and machine translation technology. Syntactic parsing technology has become a bottleneck of the current natural language processing, and hence a focus in research of computational linguistics [1]. In recent years, a large number of experts and scholars have conducted a extensive research on syntactic parsing from different angles. Qvrelid L, *et al.* [2] designed an improved data-driven dependency parsing model. Koo T, *et al.* [3] developed a third-order dependency Parser. Amati B R, *et al.* [4] studied the importance of linguistic constraints in statistical dependency parsing. Marcinczuk M, *et al.* [5] presented a named entity recognition method based on an improved hidden Markov models (HMM). Guo Hai-Xu, *et al.* [6] researched the Chinese-question syntax based on data-oriented parsing. Chen Xiao-Ming, *et al.* [7] presented a utility global optimization algorithm of PCFG syntactical parsing, which solved the implicit error in

local optimum in different stages. Zhang Feng, *et al.* [8] proposed a shallow parsing model based on CRF and transformation-based error-driven learning. Xu Jian, *et al.* [9] used HMM to build a syntax parsing system. However, these methods are all focused attention on how to construct analysis model, little literature proposed intelligent optimization algorithm to further optimize HMM model parameters for syntax parsing.

HMM is an efficient statistical method, but its greater dependence on initial values is easily trapped into local optimum trap in training. Inspired by the foregoing analysis, this paper puts forward an Adaptive Genetic Simulated Annealing HMM algorithm (or AGSA-HMM, for short) for syntax parsing. The parsing process is divided into phrase identification for short-distance structure and analytic hierarchy for long-distance structure relationship between layers in sentence. After constructing a phrase recognition HMM model, the paper combines an adaptive hybrid genetic annealing algorithm with Baum-Welch algorithm for HMM parameters' optimization training, and uses the improved Viterbi decoding algorithm to obtain phrase state sequence, and then adopts the hierarchical partitioning algorithm and Viterbi algorithm to analyze the entire sentence. Experimental results show that the AGSA-HMM method has greatly improved the performance of syntactic parsing.

2. HMM Description for Syntactic Parsing

2.1. HMM General Description for Syntactic Parsing

HMM may be regarded as a finite state automaton which generates random state transitions and outputs observation symbols. In the HMM of syntactic parsing, the observation layer is a part-of-speech sequence of phrase, and the hidden layer is a phrase state sequence. The HMM five-tuple (S, O, Π, A, B) is described as follows: ① S is phrase state set of $N=13$, including non-phrase status, noun phrase (or np, for short), verb phrase (or vp, for short), adjective phrase (or ap, for short), adverb phrases (or dp, for short), prepositions phrase (or pp, for short) quantity phrase (or mp, for short), space phrase (or sp, for short), time phrase (or tp, for short), single sentence (or dj), full sentence (or zj), and start state, end state, with a total of 13 phrase states. The 13 phrase states may be sequentially denoted as $S = \{S_0, S_1, \dots, S_{10}, S_{start}, S_{end}\}$, where the non-phrase state S_0 and S_{start}, S_{end} are all auxiliary states; ② $O = \{O_1, O_2, \dots, O_M\}$ is a symbol set composed of M word tags, the M value varies with phrase type and text structure of different length, ③ $\Pi = \{\pi_i = P(q_1 = S_i, 1 \leq i \leq 13)\}$ is the probability matrix of initial phrase state; ④ $A = \{a_{ij} = P(q_{t+1} = S_j | q_t = S_i), 1 \leq i, j \leq 13\}$ is the probability matrix of phrase state transition; ⑤ $B = \{b_j(O_k) = P(O_k \text{ at } t | q_t = S_j), 1 \leq i, j \leq 13, 1 \leq k \leq M\}$ is the phrase output probability matrix from part-of-speech group.

HMM for syntactic parsing can be described as: for each text sequence $O = w_1/p_1 w_2/p_2 \dots w_n/p_n$ with word segmentation and part-of-speech tagging, a nested phrase state sequence $S = [s_1[s_2[s_3 \dots] \dots] s_n]$ should be obtained, so that the probability $P(S/W)$ can best embody the syntactic structure of text O . This is exactly the second question of HMM: decoding problem.

2.2. HMM Topology Construction of Phrase Recognition

The goal of phrase recognition HMM is to reversely deduce the most reasonable border state sequence by part-of-speech pair sequence. The hidden state of phrase recognition is the border state between part of speech pairs. This paper sets up $N=6$ kinds of phrase border states, the transition topology of these border states is shown in Figure 1.

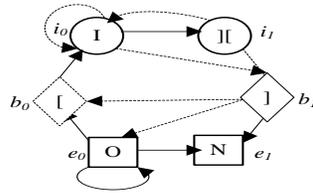


Figure 1. HMM State Transition Topology

Three kinds of geometric figures represent different states of the model and arrow represents the transition path between states in Figure 1. The square represents the possible states of phrase boundary outside: “O” for phrase outside state and “N” for one kind of non-existent state, which is used to satisfy $\sum a_{ij}=1, \sum b_j(k)=1$ in the paper. The rhombus represents phrase boundary state: “[” and “]” for phrase left and right boundary respectively. The circle represents the possible states of phrase inside: “I” for phrase inside state, “[I” for two adjacent phrase borders. Considering the late hierarchy syntactic parsing, “[I” is also viewed as a phrase inside state in the paper .

2.3. Description of Hierarchical Division Process

A kind of hierarchy combination analysis method, "divide and conquer strategy", is adopted in hierarchy division process [10], namely according to the "word → chunk → sentence” hierarchy thought, the sentence sequence is step by step bottom-up, left-right analyzed and combined on the basis of syntactic tree structure, until the whole sentence analysis is completed. While solving the phrase state sequence $S=[s_1[s_2[s_3\dots]\dots]s_n]$ in this paper, S is mapped to the following phrase state matrix to solve. Among them, S_{ij} represents the state of the i th layer and the j th part -of-speech pairs.

$$S_{\dots} = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \dots & \dots & \dots & \dots \\ S_{n1} & S_{n2} & \dots & S_{nn} \end{bmatrix}$$

In the bottom-up hierarchical parsing based on syntax tree, each layer of phrase as a whole and the same layer un-analyzed composition are combined together into higher layer of complex phrases. So S is a irregular matrix in which n decreases with different phrase composition.

3. HMM Optimization Training Based on Adaptive Genetic Annealing

Genetic Algorithm (or GA, for short) is a kind of algorithm that simulates the natural evolution law and can quickly search all solutions in solution space, but GA is prone to be premature and has slow convergence speed in the later evolutionary phase. Simulated Algorithm (or SA, for short) is a kind of stochastic optimization algorithm which is randomized to receive some inferior solutions in accordance with Metropolis criteria. SA has the features of probability leap and strong local search ability, but has the slow convergence speed and large computational quantity. Adaptive Genetic Simulated Algorithm (or AGSA, for short) improves GA and SA respectively. In the earlier stage, AGSA makes full use of the characteristics of the improved adaptive GA's global optimization and realizes the multi-population parallel search. While in the later stage, SA's Boltzmann survival mechanism is

introduced into GA to improve the local search performance, maintain population diversity, avoid premature , accelerate later convergence and improve optimization ability.

3.1. Improvement of Simulated Algorithm

The SA's search process is usually longer. In order to speed up search speed and probability of searching the global optimal solution, the cooling strategy adopts geometric cooling mechanism $T_j = \mu T_{j-1}$, ($\mu < 1$), and sets λ_{opt} and $E(\lambda_{opt})$ to be the current optimal solution and optimal value of energy function, respectively, and retains "Best so far" state. In addition, the paper also improves SA from the following aspects.

3.1.1. Selection of Energy Function: The energy function of traditional SA usually uses HMM's evaluation function $P(O|\lambda)$ as the objective function of optimization calculation. Since in hybrid model, the search target of each algorithm is different, when SA searches $E(\lambda_{opt})$ to be minimum, GA' objective function value $f(g_i)$ is maximum, HMM's likelihood probability will achieve the maximum. So the energy function is selected as follows:

$$E(\lambda) = -f(g_i) = -1 \times \frac{1}{n} \sum_{k=1}^M \ln(p(o^{(k)} | \lambda)) \quad (1)$$

3.1.2. Determination of Initial Temperature: The initial temperature of traditional SA is set according to experience with greater randomness, and the initial temperature is generally set high enough, easy to generate redundant iterations. According to the principle that initial temperature should guarantee the probability of each state in equilibrium distribution tends to equal, namely $\exp(-\Delta E_{ij}/t_0) \approx 1$, this paper improves the initial temperature as follows:

$$T_0 = (E_{min} - E_{max}) / \ln(p_0) \quad (2)$$

Where E_{max} , E_{min} and p_0 are the initial maximum energy value, minimum energy value and the worst initial acceptance probability, respectively. The initial temperature is set by adjusting p_0 lest too high initial temperature affects algorithm efficiency and too low initial temperature affects optimizing quality of solution.

3.1.3. Settings of Variable Markov Chain Length: The traditional SA stipulates temperature iterations to be a larger constant value, such that complex calculation increases. In order to make fully exchange states in annealing process, iterations should be appropriately increased with temperature reducing, to ensure a certain adaptability when temperature updates, and avoid falling into local optimum. This paper designs iterations at each temperature variable step length, *i.e.*,

$$L(T_i) = wT_i \quad (3)$$

3.1.4. Improvement of Neighborhood Solution Sampling Mechanism: Traditional SA often uses the additional perturbation method to generate neighborhood solution, and this kind of blind disturbance mechanism takes on the calculation and comparison of energy function. In this paper, the sampling mechanism to disturbance solution is improved, *i.e.*, at the time of sampling, the new algorithm locates the current solution as the origin, searches the circular area whose radius is the distance from the current solution to the nearest rectangular edge in solution space, and selects a random sample as a neighborhood solution, if better than the original solution, then instead it. The improved mechanism can accelerate SA's local optimal search.

3.2. GA's Parallel Self-adaptation

The improved model adopts the real number matrix encoding approach, a chromosome is a parameter sequence of an improved HMM, $f(g_i) = \frac{1}{n} \sum_{k=1}^M \ln(p(o^{(k)} | \lambda))$ is taken as the fitness function, n is the symbol sequence length. In addition, genetic operators are improved as follows.

3.2.1. Improvement of Selection Operators: Assuming population size is M , the improved algorithm uses the selective pressure P_s to measure the advantages of which the outstanding individual retains to the next generation, lets P_{smax} represent the largest selection pressure, generally taking 6 to 15, and lets P_{smin} represent the minimum selection pressure, generally taking 1.5 to 5, and lets G and G_{max} to be the current and maximum evolution algebra, respectively. Then

$$p_s = \frac{(G-1)(P_{smax} - P_{smin})}{G_{max} - 1} + P_{smin} \quad (4)$$

The selection probability of the k th individual is as follows:

$$p_k = \frac{2p_s(M-k) + 2(k-1)}{M(M-1)(p_s+1)} \quad (5)$$

Formula (4) is substituted into (5) to get the selection operator:

$$p_k = \frac{2(G-1)(M-k)(P_{smax} - P_{smin}) + 2(k-1)(G_{max} - 1)}{M(M-1)((G-1)(P_{smax} - P_{smin}) + G_{max} - 1)}, k = 1, 2, 3 \dots M \quad (6)$$

In the selection process, the improved algorithm first sorts population individuals by the fitness function $f(g_i)$ in descending order, and uses the strategy of preserving elite to retain high-quality individuals, then calculates the selection probabilities of all individuals according to Eq.(6). It Obviously can be seen from Eq.(6), selection operators can change dynamically with increasing evolutionary algebra, so that selection operation adapts the evolutionary process.

3.2.2. Improvement of Adaptive Crossover and Mutation Probability: Elite strategy makes the optimal individual in each generation is retained, but when the fitness achieves maximum, crossover and mutation of zero probability easily make evolution trap into local optimum and stagnate. This paper proposes an improved adaptive crossover and mutation operator.

$$p_c = \begin{cases} k_1 \exp \left[\frac{(f_{max} - f')}{f_{max} - f_{avg}} (\ln k_3 - \ln k_1) \right], & f' \geq f_{avg} \\ k_3, & f' \leq f_{avg} \end{cases} \quad (7)$$

$$p_m = \begin{cases} k_2 \exp \left[\frac{(f_{max} - f)}{f_{max} - f_{avg}} (\ln k_4 - \ln k_2) \right], & f \geq f_{avg} \\ k_4, & f \leq f_{avg} \end{cases} \quad (8)$$

Where f_{max} is the biggest fitness in population, f_{avg} is the average fitness of each generation population, f' is the larger fitness in two crossover individuals, f is the fitness of mutation individual. $0 \leq k_1, k_2, k_3, k_4 \leq 1$, $k_1 < k_2, k_3 < k_4$. The improvement of the adaptive P_c and P_m can well achieve the population diversity and the convergence of hybrid algorithm.

3.2.3. Multi-population Parallel Search and Information Exchange between Subpopulations: To effectively prevent premature, the improved algorithm divides subpopulation into I, II, III, IV four classes according to the initial fitness of individuals. Their parameter features are shown in Table 1.

Table 1. Parameter Features of Four Types of Subpopulation

subpopulation	Class I	Class II	Class III	Class IV
crossover rate	$k_1=0.1$ $k_3=0.2$	$k_1=0.2$ $k_3=0.6$	$k_1=0.4$ $k_3=0.9$	$k_1=0.7$ $k_3=1$
mutation rate	$k_2=0.05$ $k_4=0.1$	$k_2=0.1$ $k_4=0.2$	$k_2=0.2$ $k_4=0.4$	$k_2=0.3$ $k_4=0.5$
initial fitness	maximum	larger	medium	minimum

As shown in Table 1, the initial fitness of Class I is maximum, thus P_c, P_m are minimum. According to the elite strategy, Class I is called memory sub-group, in addition to its own evolution, it retains excellent individuals obtained from the other three types of evolution, which can keep the stability and diversity of individuals. The initial fitness of Class II is larger, thus P_c, P_m are relatively smaller, in addition to maintaining the individual stability, Class II can enhance the local search ability. The initial fitness of Class III and P_c, P_m are all medium. The initial fitness of Class IV is minimum, high P_c, P_m values make it easier to search for a new hyperplane, which can enhance the ability to search for the optimal solution and avoid premature.

In the course of evolution, the new algorithm regularly exchanges and updates information between sub-groups, stores the current best individual in latter three kinds of subpopulations, and updates them to the memory subpopulations, at the same time the new algorithm selects certain categories of individuals in the memory subpopulations to replace the poorer individuals in latter three kinds of subpopulations. This kind of information transfer strategy can accelerate convergence. After evolution reaches to a certain generations (multiples of G_m), all subpopulation individuals are full sorted by fitness in descending order, according to the above distribution method, individuals are reallocated to various types of subpopulations.

3.3. HMM Optimization Training Based on Adaptive Genetic Annealing

In parameter training phase, the adaptive genetic simulated annealing algorithm, AGSA, is adopted to optimize HMM initial parameters, and then Baum-Welch algorithm is used to train the optimized HMM parameters to obtain the final HMM parameters. Among them, the AGSA method is that the modified SA annealing operator is integrated into the improved adaptive GA of the multi-group parallel search to enhance optimization search performance and efficiency. The optimization training algorithm is shown in Figure 2.

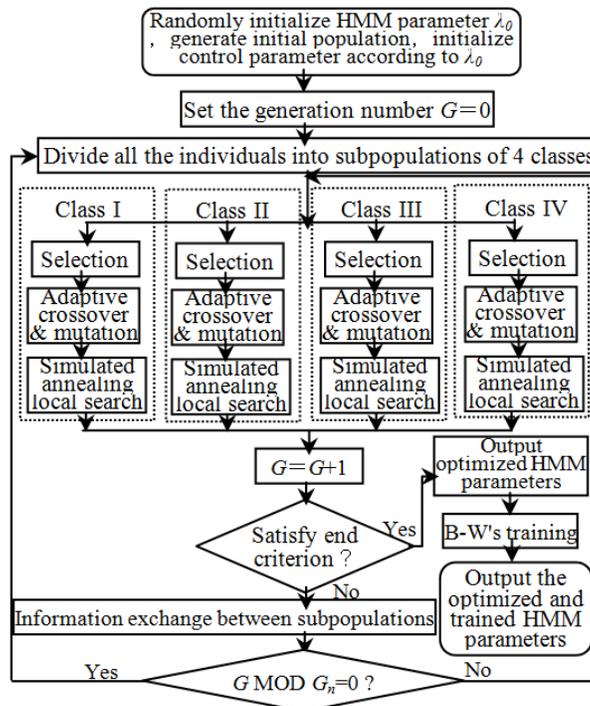


Figure 2. AGSA-HMM Optimization Training Process

Algorithm 1. AGSA-HMM optimization training algorithm

Input: training corpus with correct word segmentation and part-of-speech tagging, the HMM state number N , the initial transition probability value π , the GA population size MM , the largest evolutionary generations G_{max} , the maximum selection pressure P_{smax} and the minimum selection pressure P_{smin} , the adaptive crossover and mutation parameters k_1, k_2, k_3, k_4 , the maximum times of SA's consecutively unimproved annealing optimum value $Mmax$, the cooling coefficient μ , B-W's iterative threshold θ .

Output: the optimized and trained HMM parameter λ and likelihood probability value.

Step 1 Determine the initial value:

Randomly initialize HMM parameter λ_0 , generate initial population, initialize control parameter according to λ_0 , set the generation number $G=0$;

Step 2 Divide all the individuals into subpopulations of 4 classes in accordance with fitness;

Step 3 Realize the parallel global optimization with improved GA:

(1) Select individuals respectively from four classes subpopulations according to the selection operators in (6);

(2) Realize the adaptive crossover and mutation for four classes of subpopulations according to Formula (7),(8), respectively;

(3) Realize the simulated annealing for the each individual of four classes subpopulations, respectively, accept inferior individuals with $p = e^{-\frac{E_{old} - E_{new}}{kT}}$, update old population with new population, strengthen the local optimization search;

Step 4 Judgment of program termination criterion:

$G = G + 1$, When evolution generations $G > G_{max}$ or G exceeds a certain value M_{max} , and fitness is no longer increased, then program terminates and outputs the optimized HMM parameters, and after the training of B-W algorithm, the algorithm outputs the final optimized and trained HMM parameters and the likelihood probability value; otherwise program turns to Step 5.

Step 5 Information exchange between subpopulations:

Store the subpopulation and realize the Information exchange and update between the other three classes subpopulations;

Step 6 Reallocate the individuals of subpopulation

When evolution generations G reaches to a certain generations (multiples of G_m), program turns to Step 2, and then re-orders and reallocates according to fitness values; otherwise program turns to Step 3.

4. Syntax Parsing Process

The whole syntax analysis process is shown in Figure 3.

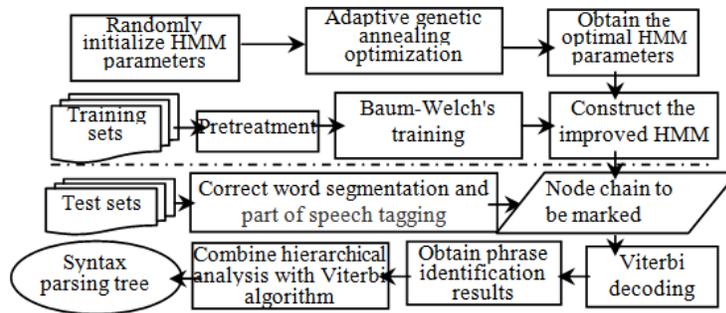


Figure 3. Syntax Parsing Process

After the training of HMM parameters, syntactic analysis is mainly composed of two processes: phrase identification and hierarchical parsing.

4.1. Phrase Identification Process

We build the adaptive genetic annealing HMM, and use the optimized HMM for identification phrase in this paper. Phrase identification process can be converted into the process of inserting different border state operators S^* . We adopt the improved Viterbi algorithm to solve the HMM decoding problem so as to obtain the state sequence S^* with the maximum $P(S|O, \lambda)$.

For example: Managers/NNS tell/VBP newcomers/NNS the/DT tale/NN of/IN their/PP countrymen/NNS./.

The final result: [zj [dj Managers/NNS [vp tell/VBP newcomers/NNS [np the/DT tale/NN of/IN [np their/PP countrymen/NNS]]]]/. .]

5. Experimental Results and Analysis

In the experiment, WSJ15-18 (altogether 8973 sentences, 21172 words and punctuation marks, about 2.18MB) of generally adapted Penn Treebank corpus are used as training set, and WSJ20 (altogether 2103 sentences,47337 words and punctuation marks) are used as test set. The test set is divided into 6 groups, 350 sentences in each group.

In order to test the effect of HMM parameter optimization training, we use the traditional Baum-Welch algorithm and AGSA presented in this paper to train HMM, respectively. The main control parameters of mixed training is as follows: GA's population size $M=200$, the maximum evolution generations $G_{max}=200$, the maximum times of consecutively unimproved annealing optimum value $M_{max}=20$, cooling coefficient $\mu=0.95$, the iterative threshold of Baum-Welch algorithm is 10^{-6} , the number of HMM states $N=13$, the initial state transition probability $\pi=[1,0,0,0,0,0,0,0,0,0,0,0]$, and A, B initial values of randomly selected. We compare the convergence of these two algorithms, the sampling error is as follows:

$$\varepsilon(p(O|\lambda)) = \sqrt{\frac{1}{N \times M} \sum_{i=1}^N \sum_{j=1}^M (\hat{p}(O|\lambda) - p(O|\lambda))^2} \tag{9}$$

Where $(\hat{p}(O|\lambda))$ is the probability in the training model, $p(O|\lambda)$ is the probability in the sample generation model ,the comparison results are shown in Figure 5.

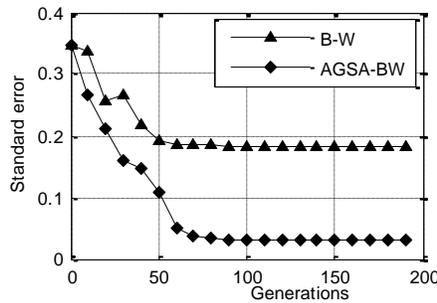


Figure 5. Standard Error Comparison of Two Algorithms

As shown in Figure 5, the standard error of B-W training algorithm begins to converge near to 0.18, while the standard error of the new algorithm begins to converge only close to 0.03, the error of the latter is reduced by about 15 percentage points compared with the former, Which proves that the new algorithm has strong search ability, convergence speed and low error and can train HMM more accurately, thereby improving system quality. Also can be seen from Figure 5, this algorithm has better algorithm stability.

To test the effectiveness of AGSA-HMM, we use Penn Treebank corpus for parsing, various phrase test results of AGSA-HMM are shown in Table 2. Wherein 6 groups of open test results of AGSA-HMM are shown in Table 3.

Table 2. Test Results of Various Phrases %

Test method		np	vp	ap	dp	pp	mp	sp	tp	dj	zj
Close test	Pre	95.51	94.21	90.91	91.89	93.2	93.64	92.19	90.91	97.08	98.09
	Rec	98.32	98.18	95.60	97.2	97.12	94.75	97.09	97.10	99.97	99.93
	F ₁	96.89	96.15	93.19	94.47	95.12	94.19	94.58	93.90	98.5	99.0
Open test	Pre	86.74	85.66	87.14	89.42	89.81	84.91	85.78	88.96	92.56	93.46
	Rec	89.78	90.09	91.07	88.73	89.10	90.04	93.19	94.18	98.97	99.31
	F ₁	88.23	87.82	89.06	89.07	89.45	87.4	89.33	91.50	95.66	96.30

Table 3. 6 Groups of Open Test Results

Test	Total sentences	Total word count	Average sentence length	Pre	Rec	F ₁
1	350	3525	11.9	95.81	96.16	95.98
2	350	3590	12.3	94.83	93.83	94.33
3	350	3555	12.4	92.77	95.92	94.32
4	350	3490	11.6	79.25	89.86	84.22
5	350	3665	12.0	81.92	87.76	84.74
6	350	3510	12.8	86.09	91.19	88.57
Average	350	3555.83	12.17	88.45	92.45	90.38

Table 2 shows that in the close test, the $F_{\beta=1}$ value of NP and VP with a higher emergence proportion is higher. This is because we use HMM statistical model and utilize AGSA to optimize HMM and make statistics information reflect more comprehensively. As a result, the more the statistics information is, the better recognition effect is. But in open test, the NP and VP recognition effect is poorer instead, it is because the nested hierarchy of NP and VP is more, resulting in the error phenomenon of phrase type and boundary location for long-distance dependencies text. These conform to the fuzzy regularity of linguistic phenomenon.

The 6 groups of open test results in Table 3 shows that their identification effects are substantially similar, which also proves the stability of the AGSA-HMM model presented in this paper.

In experiments, we compare the close test and open test results of AGSA-HMM with that of PCFG multi-stage optimization, presented by Chen Xiao-Ming, that of combination with CRF and transformation-based error-driven learning presented by Zhang Feng, etc. and that of traditional HMM presented by Xu Jian, etc., respectively. The results are shown in Table 4.

Table 4. Experimental Comparisons of Four Syntactic Parsing Methods

Method	Close test %			Open test %		
	Pre	Rec	F ₁	Pre	Rec	F ₁
CRF+ transformation	95.68	87.78	91.56	94.09	86.25	90.0
Improved PCFG	95.10	95.5	95.3	82.2	83.3	82.75
Traditional HMM	92.43	93.28	92.85	65.37	—	—
AGSA-HMM	93.76	97.53	95.61	88.45	92.45	90.38

As shown in Table 4, either in open test or in close test, the precision and recall rates of AGSA-HMM's syntactic parsing are all higher than that of traditional HMM method, which shows that the optimization effect of AGSA-HMM is obvious. Although the precision rates of AGSA-HMM are not higher than that of "CRF+ transformation" and that of "Improved PCFG" method, AGSA-HMM's $F_{\beta=1}$ value increases by about 3% on average than that of the

other three methods. These embodies the superiority of AGSA-HMM methods. Synthesizes the above data in charts and tables, we can see that the effectiveness of AGSA-HMM algorithm on syntactic parsing.

6. Conclusions

For the shortcomings of the traditional HMM training methods, we propose a HMM optimization training algorithm based on adaptive genetic annealing for syntactic parsing. The Training model uses an adaptive GA to achieve multi-population parallel search and information exchange, to control global search direction, and then utilizes a modified SA annealing operator to strengthen local optimization, which effectively overcomes premature and converges quickly with very low error. Experimental results show that AGSA-HMM's $F_{\beta=1}$ value is superior to that of several other methods, and also prove that the effectiveness and superiority of AGSA-HMM algorithm. Future researches can focus on: (1) we should develop a practical syntactic parsing system by using this model and combining with the optimal disambiguation strategy; (2) we should derive syntactic knowledge from huge quantities of unstructured text, and set up large-scale syntactic knowledge base, to further promote the research of syntactic parsing.

Acknowledgements

We would like to thank to the reviewers for their helpful comments. The Higher School Science and Technology Development Project in Shanxi Province of China (#2013147).

References

- [1] Q. L. Zhou, X. Liu, W.J. Lang and D. F. Cai, "A Divide-and-Conquer Strategy for Chunking", Journal of Chinese Information Processing, vol. 26, no. 5, (2012), pp. 120-128.
- [2] L. Qvrelid, J. Kuhn and K. Spreyer, "Improving data-driven dependency parsing using large-scale LFG grammars", Proceedings of the 47th Meeting of the Association for Computational Linguistics, Suntec, Singapore, (2009), pp. 37-40.
- [3] T. Koo and M. Collins, "Efficient Third-order dependency Parser", Proceedings of the 48th Meeting of the Association for Computational Linguistics, Uppsala, Sweden, (2010), pp. 1-11.
- [4] B. R. Amati, "Importance of Linguistic Constraints in Statistical Dependency Parsing", Proceedings of 48th Meeting of the Association for Computational Linguistics, Uppsala, Sweden, (2010), pp. 103-108.
- [5] M. Marcinczuk and M. Piasecki, "Study on named entity recognition for polish based on hidden Markov models", Proceedings of Text, Speech and Dialogue-13th International Conference, Brno, Czech Republic, (2010), pp. 142-149.
- [6] H. X. Guo, G. Y. Lu and R. Li, "Research of the Chinese-question syntax based on data-oriented parsing", Computer Engineering & Science, vol. 31, no. 10, (2009), pp. 101-104.
- [7] X. M. Chen and X. Y. Liang, "A utility global optimization algorithm of syntactical parsing PCFG", Journal of Changchun University of Science and Technology (Natural Science Edition), vol. 33, no. 2, (2010), pp. 143-146.
- [8] F. Zhang, W. G. Qu, H. Y. Zhao and J. S. Zhou, "Shallow parsing based on CRF and transformation-based error-driven learning", Journal of Guangxi Normal University, vol. 29, no. 3, (2011), pp. 147-150.
- [9] J. Xu, H. Zhang and J. S. Cai, "Building and implementation of a syntax parsing method for contemporary Chinese", Computer Applications and Software, vol. 21, no. 1, (2004), pp. 39-41, 87.
- [10] W. Z. Zhang, H. He and J. W. Ye, "A two-level cache for distributed information retrieval in search engines", The Scientific World Journal, vol. 2013, (2013), Article ID 596724, 6 pages.

Authors



Rong Li received her Master's degree in School of Computer and Information Technology from Shanxi University in Taiyuan, China in 2007. She is currently an associate professor in the Department of Computer at Xinzhou Teachers' University in Xinzhou, China. Her main research interests include intelligent information processing and machine learning. She has published over 20 research papers in scholarly journals and international conferences in the above research areas.



Hong-Bin WANG received his Ph.D. degrees in College of Information Engineering from Taiyuan University of Technology in Taiyuan, China. He is currently a professor in Xinzhou Teachers' University, China. His main research interests include intelligent information processing and information security.

