Training MEMM with PSO: A Tool for Part-of-Speech Tagging

Lei La, Qiao Guo, Qimin Cao
School of Automation, Beijing Institute of Technology, Beijing, China
lalei1984@yahoo.com.cn

***Abstract*—Maximum Entropy Markov Models (MEMM) can avoid the assumption of independence in traditional Hidden Markov Models (HMM), and thus take advantage of context information in most text mining tasks. Because the convergence rate of the classic generalized iterative scaling (GIS) algorithm is too low to be tolerated, researchers proposed a lot of improved methods such as IIS, SCGIS and LBFGS for parameters training in MEMM. However these methods sometimes do not satisfy task requirements in efficiency and robustness. This article modifies the traditional Particle Swarm Optimization (PSO) algorithm by using dynamic global mutation probability (DGMP) to solve the local optimum and infinite loops problems and use the modified PSO in MEMM for estimating the parameters. We introduce the MEMM trained by modified PSO into Chinese Part-of-Speech (POS) tagging, analysis the experimental results and find it has higher convergence rate and accuracy than traditional MEMM.**

***Index Terms*—Maximum Entropy Markov Models, Particle Swarm Optimization, dynamic global mutation probability, Part-of-Speech, text mining**

I. Introduction

Statistic models are powerful tools in many interdisciplinary associated with pattern recognition such as nature language processing (NLP), machine learning and artificial intelligence (AI).

Generally speaking, statistic models used in data mining tasks can be divided into two categories: generative models and conditional models [1]. Generative models estimate the joint probability $p(x,y)$ of the input *x* and output *y*. Hidden Markov Model is a typical generative model. HMM use the Markov chain to build associations between hidden states. However, the contextual information will be ignored because of its basic assumption: the elements of the observation sequence $(o_1 o_2 ... o_n)$ are independent of each other. On the other hand, conditional models calculate the conditional probability $p(x|y)$ and regard the outputs as dependent [2]. In this way, conditional models such as Conditional Random fields (CRFs), Markov Random Fields (MRFs), Maximum Entropy Markov Model (MEMM), etc. can improve their performance obviously by using contextual information.

Unfortunately, many conditional models such as ME and MEMM have label bias problem. Label bias refers to the ignorance of low probability events caused by the normalization of each calculation steps. An example of label bias shown as Fig. 1:

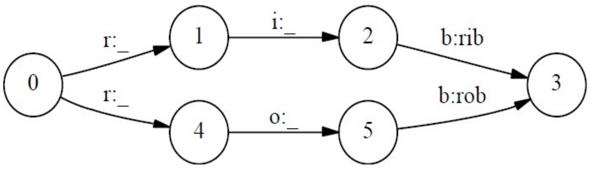


Figure 1. A example of label bias

In the most severe cases of Fig. 1, the word *rib* will be ignored because the higher probability of *rob*.

Conditional Random Fields models can solve the label bias problem ingeniously with a global normalization strategy instead of local normalization [3]. Therefore, in recent years CRFs attract wide attention of researchers and seem to become the most popular statistic models in text mining associated fields.

However, CRFs are not absolutely perfect because they need quite long training time and usually have higher computational complexity than any other types of conditional models [4]. On the other hand, MEMM is an important research field especially the aspect about how the local convergence problems of it can be solved. That means if MEMM can achieve global optimization it will be a very useful model for text mining.

Many studies focus on the Lagrange multiplier method and Newton's method to estimating the parameters of conditional models [5]. These traditional methods are theoretically wonderful but have several practical obstacles. On the contrary, intelligent optimization algorithms have a high practical value but rarely introduced into statistic models. In a word, a lot of creative works can be carried out in this field.

The first step to determine the applicability of different models in text mining is words segmentation. Furthermore, the foundation of words segmentation is Part-of-Speech (POS) tagging. Therefore, measuring the POS tagging performance of models has significance for both the selection of various text mining techniques and discussion on the development orientation of text mining theory and technology.

doi:10.4304/jsw.7.11.2511-2517

This paper is organized into six sections. After this introduction, Section II browses the important features of MEMM. After that, Section III reviews the standard Particle Swarm Optimization algorithm and modifies it by using a dynamic mutation method to achieve global optimization and log-likelihood values to prevent infinite loops. Section IV then discusses the details of training MEMM by the modified PSO algorithm. The application of MEMM trained by the modified POS is presented and analyzed in Section V with comparative data. Finally, Section VI summarizes the paper.

II. Meximum Entropy Markov Model

A. Conditional Entropy and Maximal Entropy

Conditional entropy quantifies the remaining entropy (i.e. uncertainty) of a random variable *Y* given that the value of another random variable *X* is known. It is referred to the entropy of *Y* conditional on *X*, and is written $H(Y \mid X)$. The formula as follows:

$$H(Y|X)=\sum_{x\in X}p(x)\left[-\sum_{y\in Y}p(y|x)\log_2 p(y|x)\right] \tag{1}$$

Therefore, the equation below is valid:

$$H(Y|X)=-\sum_{x\in X}\sum_{y\in Y}p(x,y)\log p(y|x) \tag{2}$$

Consider the situation that some *x* and *y* satisfy the constraint that the joint probability $p(y|x)$ is known. We can record all the constraints by some way. In practice it is common to use binary-valued trigger functions of the form:

$$f_i(x,y)=\begin{cases}1, & if\ x=x_i\ and\ y=y_i\\ 0, & otherwise\end{cases} \tag{3}$$

We can define the true expectation of *f* as $E_P(f_k)$. The empirical expectation of *f* can be calculated by summing over the training samples:

$$Ep(f)=\sum_{i=1,2,...,N}f(x_i,y_i)/N \tag{4}$$

The kernel idea of Maximal Entropy theory is divide the stochastic problem into two parts: the known part which must satisfy the constraint conditions strictly and the unknown part which must maintain the greatest uncertainty [6], in other words, maximize the entropy of the unknown part.

B. Maximum Entropy Markov Model

A Maximum Entropy Markov Model combined the Maximal Entropy theory with classic Markov model, it is a stochastic finite-state acceptor [7]. Different from Hidden Markov Model, which has both states transition and symbol emission probabilities, MEMM has only transition probabilities and the transition probabilities depend on the observations. It means that in MEMM, the states sequence is not hidden, it is a Markov chain and satisfy the constraints generated by the training examples.

Formally, a MEMM consists of a set of states $S=(s_1,s_2,...,s_n)$ and a set of transition probabilities functions A_s: $X\times S\rightarrow[0,1]$, where *X* denotes the set of all possible observations [8]. $A_S(x,s^{'})$ gives the conditional probability $P(s^{'}\mid s,x)$ of transition from s to $s^{'}$ when *x* happened. In a word, the model dose not generate *x* but only conditional on it. The principium of MEMM is shown as Fig. 2:

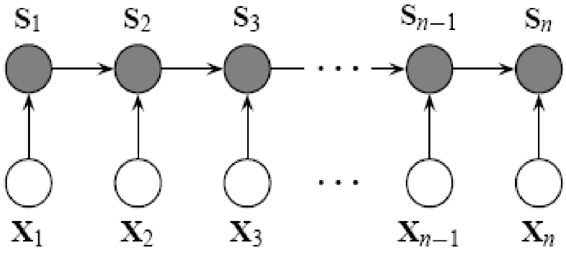


Figure 2. The principle of MEMM

For a trained MEMM, the encoding task can be solved by modifying the classic Viterbi Algorithm slightly [9]. Redefine the features function as:

$$a=\langle b,r\rangle \tag{5}$$

Where *b* is the current observation and *r* is the current state. The constraint equation becomes:

$$f\_a(o\_t,s\_t)=\begin{cases}1 & if\ b(o\_t)\ is\ true\ and\ s\_t=r\\ 0 & otherwise\end{cases} \tag{6}$$

Therefore the form of solution as:

$$P(s|\ s^{'},o)=\frac{e^{\sum_k\lambda_k f_k(x,s^{'})}}{Z(x,s)} \tag{7}$$

Where $Z(x,s)$ is a normalizing constant determined by the requirement that the sum of probability is 1.

C. Estimating Parameters of MEMM

The procedure of training MEMM is actually a process of solving the constrained optimization problem. The most general solution of this problem is using Lagrange multiplier method. This method introduces a Lagrange multiplier λ_k for every feature [10]. Define the Lagrangian $\Lambda(p,\lambda)$ by

$$\Lambda(p,\lambda)\equiv H_E(p)+\sum_k\lambda_k(Ep(f_k)-E_P(f_k)) \tag{8}$$

Scholars proposed a generalized iterative scaling (GIS) algorithm to compute the parameters of the model. The algorithm starts with an arbitrary choice of λ 's value –

for instance $\lambda_k = 1$ - for all *k*. The workflow of GIS algorithm shown as Fig. 3

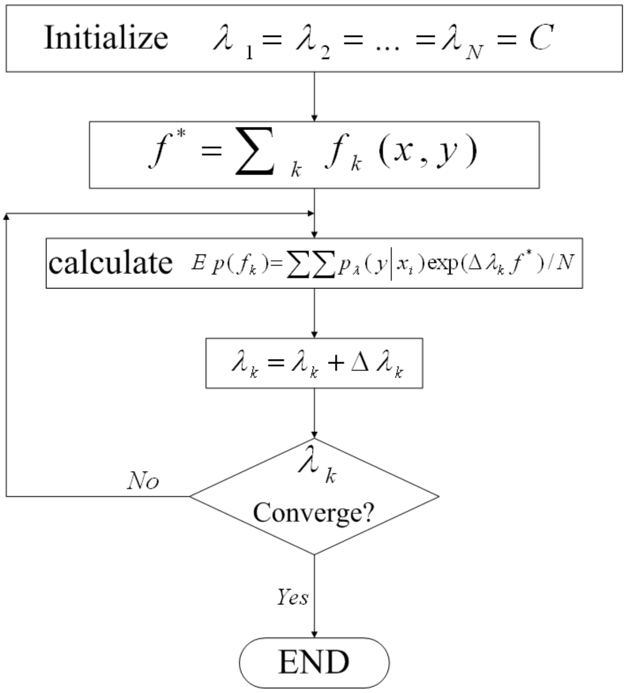

Initialize $\lambda_1 = \lambda_2 = \ldots = \lambda_N = C$
$f^* = \sum_k f_k(x, y)$
calculate $Ep(f_k) = \sum\sum p_\lambda(y|x_i)\exp(\Delta\lambda_k f^*)/N$
$\lambda_k = \lambda_k + \Delta\lambda_k$
λ_k Converge?
No
Yes
END

Figure 3. Workflow of GIS algorithm

Although the GIS algorithm has a solid theoretical foundation, it is not feasible because of its huge computational consumption in solving the exponential equations. Researchers proposed several improved algorithms to overcome the shortcomings of GIS such as Improved Iterative Scaling (IIS), SCGIS and LBFGS. However, these parameters estimation methods are not always as stable and efficient as the requirements of text mining tasks [11]. The nature of the parameters training problems is conditional optimization problem. Thus the computational swarm intelligence algorithm may solve this problem.

III. Modified Particle Swarm Optimization Algorithm

A. Particle Swarm Optimization Algorithm

In Particle Swarm Optimization algorithm, the potential solution of every optimization problem is a *particle* of the searching space. Every particle has a fitness value determined by optimized functions and a speed vector which decide its distance and orientation. In every iteration step, a particle updates itself by tracking two extreme values: the local optimal solution p_i find by itself and the global optimal solution p_g find by the entire population. Then the particle will update its position and velocity. The algorithm will repeat these steps until reach the pre-set times of iteration or lower than the required deviation. In addition, the particles' speed in all the dimensions should not beyond the max-speed set by the system [12].

The standard process for implementing PSO is shown as follows [13]:

- 1: Initialize a population array of particles with random positions and velocities on *D* dimensions in the search space.
- 2: **loop**
- 3: For each particle, evaluate the desired optimization fitness function in *D* variables.
- 4: Compare particle's fitness evaluation with $pbest_i$. If current value is better than $pbest_i$, then set $pbest_i$ equal to the current value, and $\overrightarrow{p_i}$ equal to the current location $\overrightarrow{x_i}$ in D-dimensional space.
- 5: Identify the particle in the neighborhood with the best success so far, and assign its index to the variable *g*.
- 6: Change the velocity and position of the particle according to the following equation:

$$\begin{cases} \overrightarrow{v_i} \leftarrow \overrightarrow{v_i} + \overrightarrow{U}(0,\phi_1) \otimes (\overrightarrow{p_i} - \overrightarrow{x_i}) + \overrightarrow{U}(0,\phi_2) \otimes (\overrightarrow{p_g} - \overrightarrow{x_i}) \\ \overrightarrow{x_i} \leftarrow \overrightarrow{v_i} + \overrightarrow{v_i} \end{cases}$$

- 7: If a criterion is met (usually a sufficiently good fitness or a maximum number of iterations), exit loop.
- 8: **end loop**.

The above algorithm implementation of PSO has two risks: falling into local optimization and infinite loops. These two weak points may cause the PSO could not find the best solution.

B. Dynamic Global Mutation Probability

In the later period of standard PSO, the particles swarm would convergence to local minimum or global minimum. Meanwhile p_i, p_g and x_i would point to the same value. In this situation, we need make the population take mutation to escape the local optimization and so improving the quality of solution. Mutation of the global optimized position in [14] set a threshold *T*, when the global optimized position has not be improved T times consecutively, the system would makes the p_g mutated. This method can improve the performance of PSO to some extent but maybe lead a over-modified problem because it dose not take p_g 's rate of change into account and thus the system sometime mutates too frequent to convergence.

To further enhance the accuracy and avoid the over-modified problem this paper proposed a Dynamic Global Mutation Probability (DGMP) method to modify the standard PSO algorithm.

Similar with the former global mutation method, the DGMP set a threshold Thr for mutation. However, more different from the traditional mutation method, DGMP will generate a mutation probability rather than forced to make the global optimized position mutating. In addition,

p_g 's rate of change will be introduced into the generation procedure of mutation probability. The mutation probability P_M defined as follow:

$$P_M(t+1)=\frac{1}{(1+t^{\sigma})\left|p_g(t+1)-p_g(t)\right|} \tag{9}$$

Where t is the iterate time of particle swarm, σ is the empirical constraint $0<\sigma<1$.

The system can use P_M to mutating the global optimized location and avoid mutation too frequent. Therefore the DGMP method can improve the PSO's performance and enhance the system efficiency.

C. Modify PSO with DGMP and Log-likelihood Relative Change Rate

In the former subsection, we proposed the DGMP method which can find the global optimal solutions. Note two sides of the coin that we need not only solve the local convergence problem but also avoid PSO falling into infinite loops in the optimization-neighboring area.

To prevent the endless iterations, a log-likelihood relative change-rate between iterations is used to determining whether stop the loops or not. The definition of particle swarm's log-likelihood change-rate $L(x_i)$ as follow:

$$L(x_i)=\frac{\log_2(x_i(t+1))-\log_2(x_i(t))}{\log_2(x_i(t+1))} \tag{10}$$

Where $L(x_i)$ is a threshold used to compare with a constant set by system – for instance 2×10^{-10} – to determine when jump out of the iteration [15].

Moreover, to achieve a higher performance in global searching, a linear-decreasing inertia factor $\omega(t)$ is used to combine the local searching ability with global searching ability as below [16]:

$$\omega(t)=\omega_{start}-\frac{\omega_{start}-\omega_{end}}{T_M}t \tag{11}$$

Where T_M is the maximum iteration times and t is current iteration times. The initial and final inertia factors ω_{start} and ω_{end} are empirical values between 0 and 1.

In summary, this article define the dynamic global mutation probability to determine the suitable time of mutation and thus solve local convergence problem, even more, avoid over-mutating which often happens in classic algorithms. Followed by propose DGMP, the risk of infinite loops is significantly reduced by using log-likelihood relative change rate between iterations. Finally, the inertia factor bounded by a linear-decreasing equation and empirical constants improves the global searching ability of the PSO algorithm and combined the novel methods together to construct a modified PSO algorithm with better accuracy, higher performance and lower convergence time. The detailed steps of the modified PSO algorithm as shown in Fig. 4:

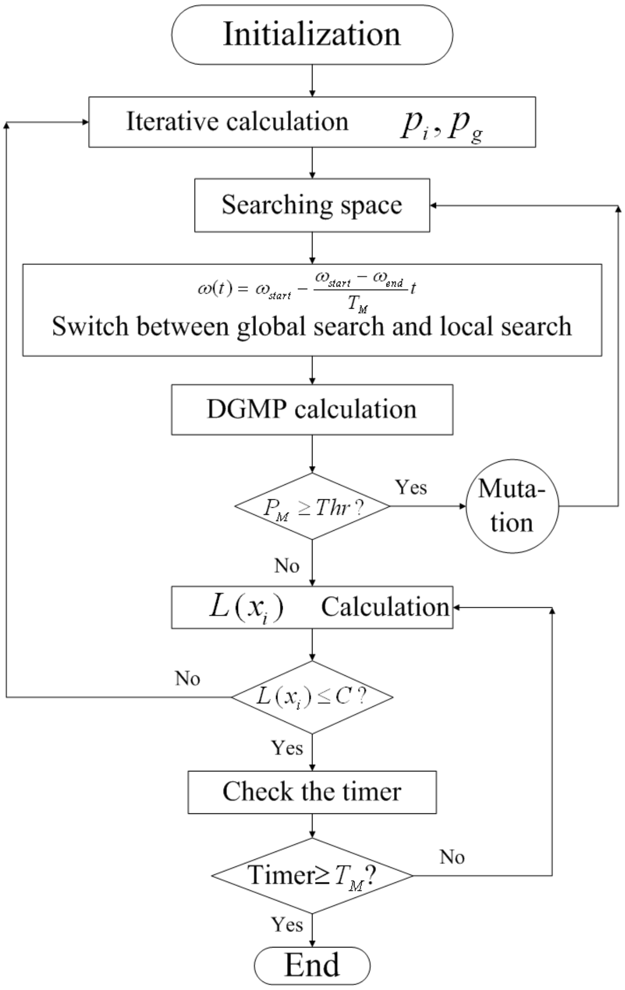


Figure 4. Detailed steps of modified PSO algorithm

IV. TRAINING MEMM WITH MODIFIED PSO ALGORITHM

A. Task Analysis

Section II explained why traditional GIS and GIS-based algorithms are inefficient and unstable. However, parameters estimation is the first step of using statistic models in text mining. We can even say that it would not be a model without appropriate parameters. Analyzing the usage of PSO in parameters estimation is the foundation of improving MEMM's utility.

Different from others algorithms which satisfy Karush-Kuhn-Tucker (KKT) theorem – those algorithms try to obtain analytical solutions through solving partial differential equations – PSO uses the local and global information of the whole population to search optimal numerical solutions.

In particular, the task in this article – training MEMM by PSO algorithm is the inverse problems of partial differential equations. Scholars had proved the existence, uniqueness and stability of solutions in MEMM modeling problems.

Therefore, in the training process, the log-likelihood equation can be used as fitness function [17] and the parameters vectors $\lambda_1, \lambda_2, ..., \lambda_N$ can be used as particle swarm. In this way, the task is transformed into computational swarm intelligence optimization problem and the parameters of MEMM can be estimated by PSO in the following subsection.

B. MEMM Parameters Estimating by Modified PSO

As the analysis above, a D-dimensional space MEMM whose number of parameters is *N* can be trained follow the steps below:

- 1: Set the current parameters – sum of particles *N*, dimension of space *D*, learning factors c_1 and c_2, inertia factor $\omega(t)$, ω_{start} and ω_{end}, the scope of possible positions S_P, the threshold of log-likelihood Thr, location of the particle *p* $L_P(i, j)$, velocity v_i, maximum iteration times T_M.
- 2: Initialize the positions and velocities of all particles in the population randomly by the functions below:

$$\begin{cases} L_p(i,j) = rand(gloS_P(j) - locS_p(j)) + locS_p(j) \\ i = 1:N, \quad j = 1:D \\ p_{i,j} = L_p(i,j), \quad p_g = 0 \end{cases} \tag{12}$$

- 3: Calculate the fitness value of every particle

$$f_{id} = \sum_{i=1}^{N} \log(p(y_i | x_i)) - \sum_{j=1}^{N} \frac{\lambda_j}{2\sigma^2} \tag{13}$$

- 4: Update the position and velocity as follow:

$$\begin{cases} v_i(t+1) = \omega(t)v_i + 2 + \frac{t}{M} rand * \\ L_p(t+1) = L_p(t) + v_i(t+1) \end{cases} \tag{14}$$

Spatially, when v_i beyond the threshold $v_{\max}$, set $v_i = v_{\max}$.

- 5: Update the DGMP

$$P_M(t+1) = \frac{1}{(1+t^{\sigma})\left|p_g(t+1) - p_g(t)\right|} \tag{15}$$

If $P_M \geq threshold$, end. Else go to step 3.

- 6: Calculate the log-likelihood $L(x_i)$, if it satisfies $L(x_i) \leq Thr$, end. Else go to step 2.
- 7: If the parameters converged or T_M reach the maximum value set by the system, end.

Until now, a novel particle swarm optimization algorithm is proposed and proved theoretically that it can be used in MEMM training to guarantee the model's practical. It called modified-PSO-based MEMM (MPSO-MEMM) in this paper.

V. Implementation and Analysis in POS Tagging

A. Chinese Part-of-Speech Tagging Task: Difficulty and Techniques

Part-of-Speech Tagging is the foundational task not only in Natural Language Process research but also in text mining applications and therefore attracted wide attention from scholars. Statisticians, computer scientists and linguists work independent or together and make notable progress in this field. However, the performance of machine tagging POS is not every single time ideal for complex corpus. Furthermore, Chinese POS tagging is more difficult because its language structure is significantly different from other languages such as Indo-European languages. In Chinese POS tagging, there is nearly no morphology changes can be used but too many disyllable words have be processed.

The major techniques used in Chinese POS tagging include HMM, second-order HMM, ME, rule-based tagging, CRFs, POS emission frequency model [18] rules and hidden Markov hybrid model, etc.

A lot of above methods have some shortcomings such as high computational complexity, low precious, lack of robustness, long training time, low degree of automation and so on. On the other hand, even MEMM is a will developed model, Chinese POS based on MEMM is still an open issue because there have no adequate literatures can be used as reference.

B. MPSO-MEMM Modeling

The experiment use Modern Chinese Corpus provided by Center of Chinese Linguistics, Peking University as its training set and the notification text in Ministry of Education of the People's Republic of China' website http://www.moe.edu.cn/publicfiles/business/htmlfiles/moe/moe_0/index.html randomly selected from January 30th, 2011 to July 30th, 2011 as its test set.

Before the POS tagging, the pretreatment was processed for training corpus and testing corpus according to the corpus annotation in order to prevent the interference of factors which do not expected.

Like [19], we use a semi-automatic approach for feature selection. Features are obtained by two steps, the first of which is to establish feature templates, and the second is to extract features from training corpus according to the feature templates.

Smoothing algorithm is utilized to the transition probability in tag bi-gram model. Because not all the POS tags can transfer between each other, three transition restricted rules are used to reduce the sum of tag pairs. It can make smoothing more reliable. Let *X* be a certain POS type and *Y* be a random POS type.

- *B-X* can be followed by *I-X*, *B-Y* or *O*.
- *I-X* can be followed by *B-Y*, *I-X* or *O*.

- *O* can be followed by *B-Y*, or *O*.

Through the three rules above, 298 types of pairs can be enumerated. Interpolation smoothing is used and the smoothing formula can be defined as

$$P(s_i \left| s_{i-1} \right.) = \varepsilon \cdot p'(s_i \left| s_{i-1} \right.) + (1-\varepsilon) \cdot p(s_i) \quad (16)$$

Set empirical value ε as 0.69.

We randomly select a set contains 1.5 million Chinese characters as the training samples. The training time is nearly 11 hours (652 minutes). The comparison of training time of ME, MEMM, MPSO-MEMM and CRFs with different size [20] of training sets is shown in TABLE I.

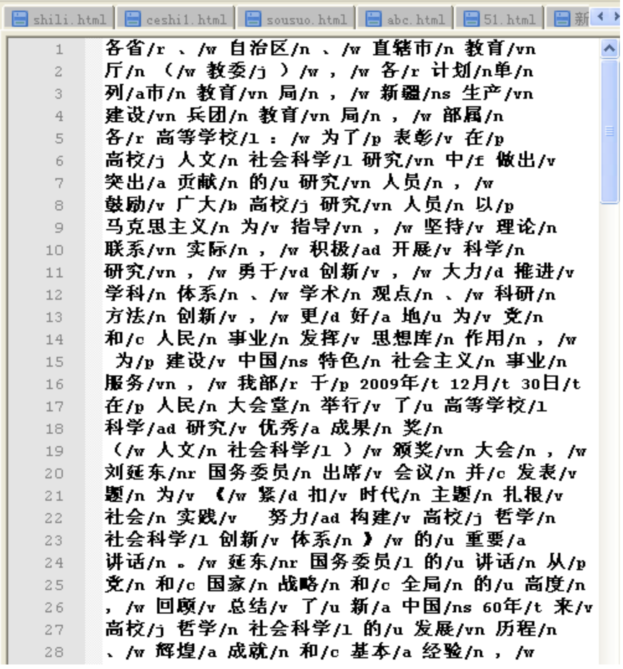
TABLE I.
TRAIN TIME OF ME, MEMM AND MPSO-MEMM

Models Size	ME	MEMM	CRFs	MPSO-MEMM
15K	6min	7min	9min	4min
150K	73min	89min	103min	26min
1.5M	11h	18h	Too long!	7h

The estimation efficiency of MPSO-MEMM is so proved higher than traditional MEMM and ME. It is interesting to note that, verified by Shen [21], when the training sets are larger than 1 million Chinese characters, the procedure of CRFs estimation is more than a day, which seems endless and need use work station instead of common PC to run the algorithm.

C. POS Tagging Experiment and Analysis

The experiment based on open source project *mallet* (machine learning for language toolkit). Detail information can be found in its website [22].



```
各省/r 、/w 自治区/n 、/w 直辖市/n 教育/vn
厅/n （/w 教委/j ）/w ，/w 各/r 计划/n单/n
列/a市/n 教育/vn 局/n ，/w 新疆/ns 生产/vn
建设/vn 兵团/n 教育/vn 局/n ，/w 部属/n
各/r 高等学校/l ：/w 为了/p 表彰/v 在/p
高校/j 人文/n 社会科学/l 研究/vn 中/f 做出/v
突出/a 贡献/n 的/u 研究/vn 人员/n ，/w
鼓励/v 广大/b 高校/j 研究/vn 人员/n 以/p
马克思主义/n 为/v 指导/vn ，/w 坚持/v 理论/n
联系/vn 实际/n ，/w 积极/ad 开展/v 科学/n
研究/vn ，/w 勇于/vd 创新/v ，/w 大力/d 推进/v
学科/n 体系/n 、/w 学术/n 观点/n 、/w 科研/n
方法/n 创新/v ，/w 更/d 好/a 地/u 为/v 党/n
和/c 人民/n 事业/n 发挥/v 思想库/n 作用/n ，/w
 为/p 建设/v 中国/ns 特色/n 社会主义/n 事业/n
服务/vn ，/w 我部/r 于/p 2009年/t 12月/t 30日/t
在/p 人民/n 大会堂/n 举行/v 了/u 高等学校/l
科学/ad 研究/v 优秀/a 成果/n 奖/n
（/w 人文/n 社会科学/l ）/w 颁奖/vn 大会/n ，/w
刘延东/nr 国务委员/n 出席/v 会议/n 并/c 发表/v
题/n 为/v 《/w 紧/d 扣/v 时代/n 主题/n 扎根/v
社会/n 实践/v   努力/ad 构建/v 高校/j 哲学/n
社会科学/l 创新/v 体系/n 》/w 的/u 重要/a
讲话/n 。/w 延东/nr 国务委员/l 的/u 讲话/n 从/p
党/n 和/c 国家/n 战略/n 和/c 全局/n 的/u 高度/n
，/w 回顾/v 总结/v 了/u 新/a 中国/ns 60年/t 来/v
高校/j 哲学/n 社会科学/l 的/u 发展/vn 历程/n
、/w 辉煌/a 成就/n 和/c 基本/a 经验/n ，/w
```

Figure 5. Part-of-Speech tagging using MEMM

We randomly download notification texts totally contain 20730 Chinese characters from the website of Ministry of Education of the People's Republic of China as the test corpus. The experimental performance as shown in Fig. 5

The test set has a copy which been pre-tagged carefully by graduate major in modern Chinese as a criterion used to evaluate the experimental data.

The POS tagging result of MEMM in this article is 18121 characters of the test set – a total of 20730 Chinese characters – are tagged accurately. The Precision of the model can be calculated as:

$$\text{Precision} = \frac{rightly}{sum} \times 100\% = \frac{18121}{20730} \quad (17)$$

Thus the PSO tagging precision of MEMM in this article is 87.4%. Performances of different models are compared in Fig. 6:

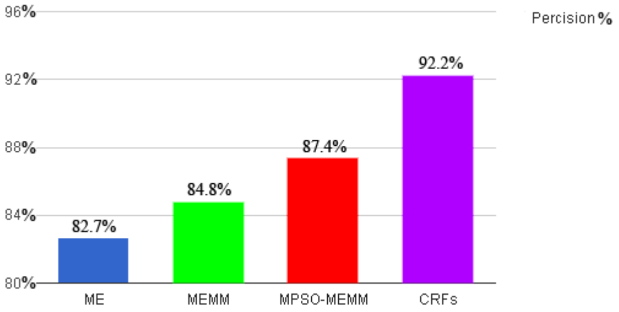


Figure 6. PSO performance of different models

The MPSO-MEMM has better performance than ME and traditional MEMM but less precision than CRFs through comparing with other scholars' research. However, CRFs sometimes need a terrible training time and have huge computational complexity. Therefore, research on how to improve MEMM has practical values to enhance Part-of-Speech precision with reasonable computational complexity and time consumption.

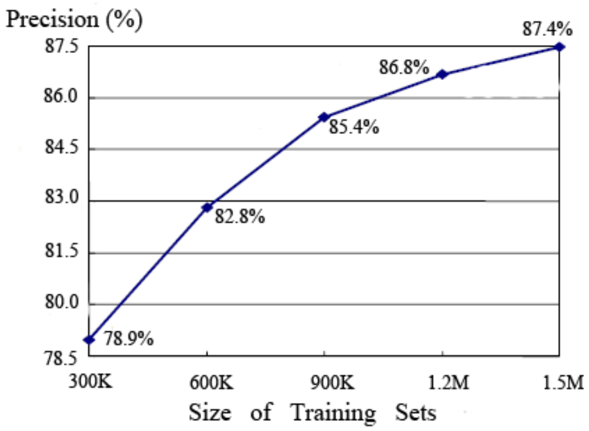


Figure 7. Relation of precision and training sets size.

The experiment reveals that MPSO-MEMM is unfortunately high corpus dependant like other statistic models. That means higher precision need larger training set. Relation between accuracy and training sets size as shown in Fig. 7. This relationship means we must pay more for get better performance.

VI. CONCLUSION

Maximum Entropy Markov Model is a method with very solid theoretical foundation but its utility in NLP tasks is quite limited because the long convergence time and label bias problems. This article proposed a novel mutation tool－Dynamic Global Mutation Probability to avoid local convergence in standard Particle Swarm Optimization algorithm and use the improved PSO algorithm in parameters training of MEMM. Therefore a modified PSO-based MEMM model is constructed to overcome the drawbacks of traditional MEMM. Experimental results show that the MPSO-MEMM has higher training efficiency and accuracy than classic ME and MEMM models. Moreover, it has lower computational consumption and it is more robust than MEMM. Although its accuracy is not as good as CRFs, the higher efficiency makes it wealthy of get more attention in further improving and applications.

Other intelligence optimization algorithms such as GE, SA and ACO may also get good performance in training MEMM. The MEMM's estimation time should be further reduced and the precision should be further improved in order to compete with CRFs. In addition, the applications of MPSO-MEMM in other text mining tasks such as categorization, clustering, information extraction, etc. will probably benefit for researchers and related businesses. These will be undertaken as future works on this topic.

REFERENCES

[1] D. Hewlett and P. Cohen, "Word segmentation as general chunking", *Psychocomputational Models of Language Acquisition Workshop (PsychoCompLA)*, July 29, 2009.

[2] Yoong Keok Lee, Aria Haghighi, and Regina Barzilay, "Simple type-level unsupervised POS tagging", *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, Cambridge, MA, October, 2010.

[3] John Lafferty, Andrew McCallum, and Fernando Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", Proceedings of ICML 2001.

[4] Tao Dong, Wenqian Shang and Haibin Zhu, "An Improved Algorithm of Bayesian Text Categorization", *Journal of Software*, VOL. 6, September, 2011.

[5] Li and M. Sun, "Punctuation as implicit annotations for Chinese word segmentation", *Computational Linguistics*, 35. 2009.

[6] Paul Cohen, Niall Adams, and Brent Heeringa, "Voting Experts: An Unsupervised Algorithm for Segmenting Sequences", *Intelligent Data Analysis*, VOL. 11, December, 2007.

[7] Adnan Abdul-Aziz Gutub, Fahd Al-Haidari, Khalid M Al-Kahsah and Jamil Hamodi, "e-Text Watermarking: Utilizing 'Kashida' Extensions in Arabic Language Electronic Writing", *Journal of Emerging Technologies in Web Intelligence*, VOL 2, 2010.

[8] Jian Wang, Wenwu Shao and Fei Zhu, "Biological Terms Boundary Identification by Maximum Entropy Model", *6th IEEE Conference on Industrial Electronics and Applications*, 2011.

[9] Hong-Kwang and Jeff Kuo, "Maximum entropy modeling for speech recognition", *2004 International Conference on Chinese Spoken Language Processing*, ISBN: 0-7803-8678-7, December, 2004.

[10] Li Rong, Liu Li-ying and Fu He-fang, "Application Study of Hidden Markov Model and Maximum Entropy in Text Information Extraction", *International Conference on Artificial Intelligence and Computational Intelligence* November 07-08, 2009.

[11] Ciuperca Gabriela, Girardin Valerie and Lhote Loick, "Computation and Estimation of Generalized Entropy Rates for Denumerable Markov Chains", *IEEE Transactions on Information Theory*, VOL 57, August, 2011.

[12] SUN Jun, LIU Jing and XU Wenbo, "Using quantum-behaved particle swarm optimization algorithm to solve non-linear programming problems", *International Journal of Computer Mathematics*, *Distributed Algorithms in Science and Engineering*, 2007.

[13] JIN Yixiong, CHENG Haozhong and YAN Jianyong, "Improved particle swarm optimization method and its application in power transmission network planning", *Proceeding of the CSEE*, 2005.

[14] Shirazi Masoud Jahromi, Vatankhah Ramin and Boroushaki Mehrdad, "Application of particle swarm optimization in chaos synchronization in noisy environment in presence of unknown parameter uncertainty", *Communication in Nonlinear Science and Numerical Simulation*, December, 2011.

[15] Lin Lu, Qi Luo, Jun-yong Liu and Chuan Long, "Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients", *IEEE Transactions on Evolutionary Computation*, October, 2004.

[16] Cervantes A, Galvan I.M and Isasi P, "AMPSO: A New Particle Swarm Method for Nearest Neighborhood Classification", *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, VOL 39, 2009.

[17] Zexuan Zhu, Jiarui Zhou and Zhen Ji, "DNA Sequence Compression Using Adaptive Particle Swarm Optimization-Based Memetic Algorithm", *IEEE Transactions on Evolutionary Computation*, VOL 15, August, 2011.

[18] Ting Liu, Wanxiang Che and Sheng Li, "Semantic Role Labeling with Maximum Entropy Classifier" *Journal of Software*, Vol.18, No.3, March 2007.

[19] Zengfa Dou and Lin Gao, "A Bio-Entity Recognition Algorithm for Literature by Conditional Random Field Model Based on Improved Particle Swarm Optimizer", *Journal of Xi'an Jiaotong University* VOL 44, December, 2010.

[20] Tomas. Pedersen. "The effect of different context representations on word sense discrimination in biomedical texts", *Proceedings of the 1st ACM International IHI Symposium*, 2010.

[21] Guoyang Shen, Likun Qiu, Changjian Hu and Kai Zhao, "CCRFs: Cascaded Conditional Random Fields for Chinese POS Tagging", *International Conference on Natural Language Processing and Knowledge Engineering, (NLP-KE)*, 2009.

[22] http://mallet.cs.umass.edu/index.php.