TWO-STAGE MODEL SELECTION WITH PARAMETERS WEIGHTED HIDDEN MARKOV MODELS AND LIKELIHOOD RATIO FOR PART-OF-SPEECH TAGGING

Shichang Sun[*], *Hongbo Liu*[†], *Pixi Zhao*[‡], *Hongfei Lin*[§]

Abstract: In many natural language processing applications two or more models usually have to be involved for accuracy. But it is difficult for minor models, such as "backoff" taggers in part-of-speech tagging, to cooperate smoothly with the major probabilistic model. We introduce a two-stage approach for model selection between hidden Markov models and other minor models. In the first stage, the major model is extended to give a set of candidates for model selection. Parameters weighted hidden Markov model is presented using weighted ratio to create the candidate set. In the second stage, heuristic rules and features are used as evaluation functions to give extra scores to candidates in the set. Such scores are calculated using a diagnostic likelihood ratio test based on sensitivity and specificity criteria. The selection procedure can be fulfilled using swarm optimization technique. Experiment results on public tagging data sets show the applicability of the proposed approach.

Key words: *Hidden Markov models, model selection, part-of-speech tagging, likelihood ratio*

Received: February 20, 2012
Revised and accepted: June 8, 2012

[*]Shichang Sun
School of Computer, Dalian University of Technology, 116024 Dalian, China, with an affiliate appointment in the School of Computer, Dalian Nationalities University, 116600 Dalian, China, E-mail: schsunster@gmail.com

[†]Hongbo Liu -Corresponding Author
School of Information at Dalian Maritime University, 116026 Dalian, China, with an affiliate appointment in the School of Computer, Dalian University of Technology, 116024 Dalian, China, E-mail: lhb@dlut.edu.cn

[‡]Pixi Zhao
School of Computer, Dalian University of Technology, 116024 Dalian, China, with an affiliate appointment in the School of Computer, Dalian Nationalities University, 116600 Dalian, China, E-mail: pxzh@dlnu.edu.cn

[§]Hongfei Lin
School of Computer, Dalian University of Technology, 116024 Dalian, China, E-mail: hflin@dlut.edu.cn

1. Introduction

Hidden Markov Models (HMM) have found many successful applications in Natural Language Processing (NLP) areas [1, 2, 3], and model selection with HMMs is widely used to achieve better performance [4, 5, 6]. GRAMOFON [7] uses model-network for the selection of an appropriate subset of models for the case that different models share similar error characteristics. Chien and Furui [8] develop a predictive information criterion to estimate HMM and simultaneously select the proper size of HMM to represent the observed data. HMM model selection is useful not only for the adaptation of HMM parameters [9], but also for the representation of n-gram language model [10]. Full model selection (FMS) problem is defined in [11] as selecting; the combination of a pool of processing methods, feature selection and learning algorithms that obtain the lowest classification error for a given data set. Its task also includes the selection of hyperparameters for the considered methods. In FMS both the model parameters and hyperparameters are optimized simultaneously according to the same training set. In many applications including part-of-speech (POS) tagging, however, each model as a candidate for model selection may have different function and emphasis. A few major models use many random variables to model context relationship and conditional independence structure, whereas other minor models include heuristic rules and features. Although minor models may not be able to handle the whole recognition problem, they sometimes accommodate the experiences of field experts in a direct and easy way. It is usually difficult for the major probabilistic model and minor models to work together smoothly. We present a method to exploit the potential performance of the major models in POS, i.e. HMM, by overcoming the limitation of training corpus with the help of minor models.

In many applications, including POS, HMM is a major model. The delicate structures make HMM capable of classifying complex and structured objects in sequence recognition problems. The parameters of HMM include transition probabilities and emission probabilities, both of which are learned from labeled training data. As in many machine learning problems, however, the robustness of model is hindered by the lack of sufficient labeled training data. Freitag [12] uses a statistical technique called "shrinkage" for smoothing in parameter estimation. Self-adaptive design approach [13] focuses on learning the correct states number with a self-adaptive procedure. The design of [14] aims to fit the length distribution of sequences. Although above methods have made progressive effort to balance the delicacy of structures and the robustness of parameters estimation in many fields, such balanced models still bear parameters uncertainty because in practice the trained model parameters are limited by the sparseness and noisiness of training data.

Theoretically, HMM is faster than exponential models such as conditional random fields (CRF), but classically HMM does not provide an easy way to use features. Besides, as a statistical model, HMM has only one set of determined parameters and is hard to deal with the influence of sparseness and noisiness of training data. We try to increase the robustness by extending learned parameters into intervals and to improve the performance by introducing model selection with likelihood ratio. We propose a two-stage approach for model selection between hidden

Markov model and other minor models that use heuristic rules and features. In the first stage, the major model is extended to give a set of candidates for model selection. Parameters Weighted Hidden Markov Model (PWHMM) is presented using weighted ratio to create a set of HMM models which can be viewed intuitively as learned from different part of training data that emphasize either the transition probabilities or the emission probabilities. In the second stage, heuristic rules and features are used as minor models to give extra scores to candidates in the set. Such scores are calculated using a diagnostic likelihood ratio test. The selection procedure can be fulfilled using optimization techniques such as swarm intelligence.

The remainder of this paper is organized as follows. In Section 2 we review the background and related work. In Section 3 we present the details of the two-stage HMM model selection strategy. Next, we present our experimental evaluation of our method for POS tagging problem in Section 4, followed by the conclusions in Section 5.

2. Background and Related Work

POS tagging is considered a fundamental part of natural language processing, which aims to computationally determine a POS tag for a token in text context. POS tagger is a useful preprocessing tool in many NLP applications such as information extraction and information retrieval [1, 2, 15, 16, 17, 18].

POS tagging problem has been modeled with many machine learning techniques, which include hidden Markov models [1], maximum entropy models [19], support vector machines, conditional random fields [20], etc. Each model can have good performance after careful adjustment such as feature selection, but HMMs have the advantages of small amount of data calculation and simplicity of modeling. In [15], HMMs combined with good smoothing techniques and with handling of unknown words work better than other models. For such a sequence recognition problem, the classical EM algorithms and Viterbi algorithms for HMM can be found in [21, 22, 23, 24].

In part-of-speech tagging and many other applications in NLP, there are many minor models called "backoff" tagger, which means the major tagger leaves "not-sure" tokens blank and pass the tagging work to the "backoff" tagger [25]. But the most widely used HMM state sequence recognition criterion is to find the *single* best state sequence, so the "backoff" taggers are not suitable to use. Besides, the usage of "backoff" taggers may lose the context dependent information and harms the effort of probabilistic models on capturing context relationship to achieve an accurate tagger. The proposed approach chooses to output the whole sequences produced by probabilistic models and simultaneously to overcome the intrinsic disadvantages of probabilistic models on overtraining and data sparseness with the adjustment by minor models.

HMM is a probabilistic model for modeling time series data. It extends the concept of Markov random process to include the case where the observation is a probabilistic function of the states. Thus, HMM is a double stochastic process that allows a flexible layer of random variables for a large amount of observable events. One of them is hidden state which is not directly visible, and each state can emit observable output symbols determined by its own probability distribution.

This extension makes HMM applicable to many fields of interest such as NLP, where the amount of observable events, i.e. words, is often as big as hundreds of thousands [21]. However, this delicacy of structures leaves the robustness of the model easily influenced by the bias of the training set.

The potential performance of HMM is limited because it is traditionally based on pre-determined model parameters. By randomizing the learned model parameters, there is room for HMM algorithms to perform better. The methods in [26, 27] need the representation of fuzzy relationship or fuzzy rules and are less cost-effective than the proposed work. Our approach is more specific to HMM structure, i.e. directly using the ratio between transition probabilities and emission probabilities. Zeng [28] presents a fuzzy-set method to allow randomness in HMM and achieves robust performance on speech variation, but a Gaussian primary membership function has to be used for each state. In our study, the ratio between transition parameters and emission parameters is weighted to adjust the bias of HMM. The sensitivity and specificity is used by Li [29] only as optimization criteria for support vector machine model selection. We use a diagnostic likelihood ratio test to enable the help of experience-rules and features that are difficult to use in HMM.

3. Model Selection Strategy

We propose a two-stage approach to exploit the potential performance of HMM by overcoming the limitation of training corpus with the help of minor models. As in Fig. 1, in the first stage PWHMM produces a set of HMM models, in the second stage minor models can give more extra scores to promising candidates in the set. Such scores are calculated using a diagnostic likelihood ratio test.

First we present the specification of PWHMM. Afterward we discuss the usage of diagnostic likelihood ratio for model selection in Section 3.2 and provide an algorithms to search the best model in the set in Section 3.3.

3.1 Parameters Weighted Hidden Markov Model

To create a set of HMM candidates around the HMM learned through dataset, the Parameters Weighted Hidden Markov Model (PWHMM) is presented using weighted ratio. Such HMM set can be viewed intuitively as learned from different part of training data that emphasize either the transition probabilities or the emission probabilities.

For clarity purposes, the specification of PWHMM is presented based on the classical HMM [21] as follows:

- States
  $S = \{S_1, \cdots, S_N\}$ denotes the hidden state set, N represents the number of these states. In POS tagging problem, S stands for the part-of-speech tags. The part-of-speech tags carry structural significance although they are hidden in human language.

- Outputs
  $V = \{v_1, \cdots, v_M\}$ denotes the set of output symbols produced by states, M

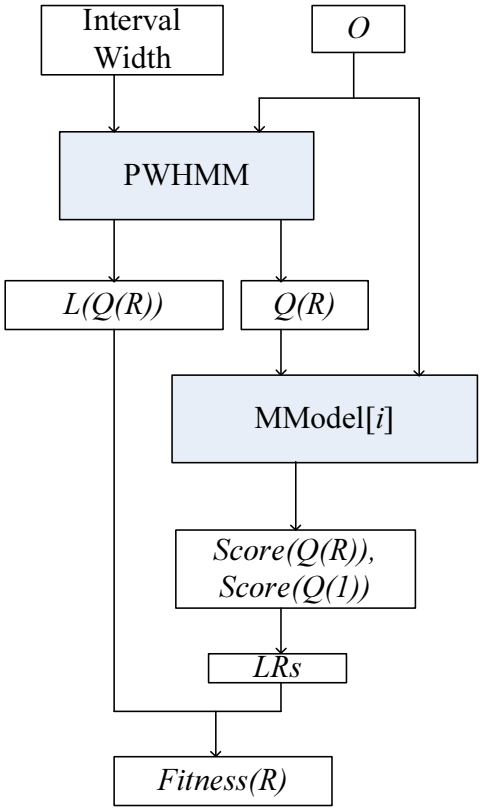

Fig. 1 *The block diagram of the two-stage method.*

represents the number of these symbols. In POS tagging problem, V stands for vocabulary of the language and M is the alphabet size.

- Transition probability matrix
  $A = a_{ij}$ denotes the state transition matrix, where $a_{ij} = P[q_{t+1} = S_j | q_t = S_i]$, $1 \leq i, j \leq N$. In POS tagging problem, A depicts the statistical frequency of the transitions between part-of-speech tags.

- Observation symbols emission distribution
  $B = b_j(k)$ denotes the observation symbols emission distribution, where $b_j(k) = P[V_k \ at \ t | q_t = S_j]$, $1 \leq j \leq N$, $1 \leq k \leq M$. In POS tagging problem, B depicts the statistical frequency of words being categorized into some part-of-speech tags.

- Initial states distribution
  Π denotes the initial states distribution. It is a vector of initial states, where $\Pi_i = P[q_1 = S_i]$, $1 \leq i \leq N$. Where q_1 is the state at initial time that satisfies two constraints $0 \leq \Pi_i \leq 1$, $1 \leq i \leq N$, and $\sum_{i=1}^{N}(\Pi_i) = 1$.

- Observation Sequence
  Observation Sequence is a sequence of tokens to recognize. O denotes the observation sequence and $O = O_1 O_2 \dots O_T$, and T is the length of the sequence. In POS tagging problem, O is the sentence in the target language.

- Ratio Weighting
  The Ratio Weighting R is defined to adjust the ratio between the transition probabilities and the emission probabilities.

- Interval Width
  Interval Width is the effective and efficient range for Ratio Weighting R. This interval is chosen to be symmetrically around 1 and is determined experimentally as in this paper.

- State Sequence Set
  State Sequence Set is a set of states sequence produced with Ratio Weighting R. $Q(R)$ denotes the state sequence set. In POS tagging problem, $Q = q_1 q_2 ... q_T$ stands the labeled tags.

- Sequence Value
  Given an observation sequence, sequence value is the log-likelihood value of a state sequence produced by HMM Viterbi algorithm. The sum of logarithms likelihood probabilistic value along the sequence $Q(R)$ is denoted by $L(Q(R))$.

We present W-Viterbi algorithm in Algorithm 1 for model expansion. According to [21], the most widely used state sequence recognition criterion is to find the *single* best state sequence by the following equation.

$$\delta_t(i) = \max_{q_1, \cdots, q_{t-1}} P[q_1 \cdots q_t = i, o_1 \cdots o_t | \lambda]. \tag{1}$$

Given a Ratio Weighting R, W-Viterbi algorithm can recognize a sequence of tokens by solving Eq. (2), where $\delta_t(j)$ is the best score along a single path at time t which accounts for the first t observations and ends in state S_j. Our proposed method is basically a dynamic programming method to find the optimal state sequence associated with the given observation sequence.

$$\delta_t(j) = \max_{1 \le i \le N} [\delta_{t-1}(i) a_{ij}^R] b_j(O_t). \tag{2}$$

From the view of dynamic programming, the value function (2) consists of two parts, the vertex value and the node value. So the Ratio Weighting R in PWHMM can be viewed as a ratio between the vertex value and the node value in dynamic programming. From the experimental evaluation, we will show that better candidate models can be produced by the adjustment of this ratio. So this method can be extended to other dynamic programming problems.

3.2 Minor models used as likelihood ratio

In the second stage, minor models are used as evaluation functions to act as extra scores for HMM candidate. Minor models such as "backoff" taggers can be easily

Algorithm 1 Weighted-Viterbi Algorithm

Input: A sequence of tokens O, Ratio Weighting R
Require: Transition probability matrix A, Observation symbols emission distribution B, Initial states distribution Π
Output: A sequence of tags $Q(R)$
01. **Initialization:**
02. $\delta_1(i) = \Pi_i b_i(o_1)$, $1 \leq i \leq N$
03. $\psi_1(i) = 0$ // an array to store best states
04. **for** $2 \leq t \leq T$, $1 \leq j \leq N$:
05. **update** $\delta_t(j)$ according to Eq. (2)
06. $\psi_t(j) = argmax_{1 \leq i \leq N}[\delta_{t-1}(i)a_{ij}]$
07. **Termination:**
08. Set $q_t* = argmax_{1 \leq i \leq N}[\delta_T(i)]$
09. $L = log(\max_{1 \leq i \leq N}[\delta_T(i)])$ - differences caused by R in Eq. (2)
10. Store state sequence from $t = T - 1$ to 1: $q_t* = \psi_{t+1}(q_{t+1}*)$
11. return $Q = q_1 q_2 ... q_T$

coded into evaluation functions. Affix tagger is one of the "backoff" taggers that use heuristic rules based on affix (and suffix) of a token to produce a tag. It can be coded into an evaluation function that counts the number of *right* token-tag pairs under the rules. Another evaluation function, Unknown-pairs, counts the number of token-tag pairs that did not appear in the training set.

Such scores are calculated using a diagnostic likelihood ratio test based on sensitivity and specificity criteria. The calculated likelihood ratio provides a direct estimate of how much a test result will change the possibility of being a good solution. The possibility of $Q(R)$ as a good solution is measured using the HMM-log-likelihood.

The likelihood ratios LRs are calculated using Tab. I and Eq. (3). The test is specific to the candidate solutions $Q(R)$ and the HMM-produced solution $Q(1)$. The test result *positive* means the evaluation function scores more on $Q(R)$ than $Q(1)$, and *negative* means the inverse. The fact *better* indicates $Q(R)$ is a good solution (better than the HMM-produced solution), whereas the *worse* indicates the opposite fact. The number of occurrences of above conditions are counted as a, b, c, d as in Tab. I. The likelihood ratio is calculated using the sensitivity and specificity of the test as in Eq. (3). Sensitivity is the proportion of actual positives which are correctly identified as such. Specificity measures the proportion of negatives which are correctly identified. An applicable evaluation function should have reasonable sensitivity and specificity values. So the sensitivity and specificity criteria can be used to test if a minor model is helpful.

$$\begin{aligned} sensitivity &= a/(a+b) \\ specificity &= d/(c+d) \\ LR(+) &= sensitivity/(1-specificity) \\ LR(-) &= (1-sensitivity)/specificity \end{aligned} \tag{3}$$

Likelihood ratios of a minor model are used to adjust the logarithmic likelihood produced by HMM as in Eq. (4) when the test is positive, and as in Eq. (5) when

	positive	*negative*
better	a	b
worse	c	d

Tab. I *Sensitivity and Specificity Calculation.*

the test is negative. Since we use log-likelihood in HMM, the logarithm values are used for LRs. To overcome the imbalance of the numbers positive features and negative features, we use the mean value of logarithmic likelihood ratios for each minor model.

$$L = L + (log(LR(+)) - log(LR(-)))/2 \tag{4}$$

$$L = L - log(LR(+)) - log(LR(-)))/2 \tag{5}$$

3.3 Model selection algorithm

We first define some terminologies for model selection between PWHMM and minor models.

- Minor Models Vector
  Minor Models $MModels$ are a set of heuristic rules or features that can be coded into evaluation functions. $Score(Q)$ denotes the score by such associated evaluation functions on State Sequence Qs given by the Observation Sequence O and the training dataset.

- Likelihood Ratio Vector
  The Likelihood Ratio Vector is the result of likelihood ratio test for minor models, and is denoted by LRs. LRs is calculated as in section 3.2 by a function using Minor Models Vector, the training corpus and the sequences given by the Ratio Interval.

- Ratio Weighting Fitness
  The Ratio Weighting Fitness is the fitness value for a given Ratio Weighting R, and is denoted as $Fitness(R)$. It is used in the proposed two-stage model selection to search for best state sequence by applying the LRs to L.

The algorithm for finding the best state sequence in our model selection method can now be stated in Algorithm 2. $Q(R)$ and $L(R)$ are calculated in line 5. Then, the likelihood ratio test is performed (lines 7-10). Lines 11-13 calculate the fitness and choose the solution of best model with biggest fitness.

The object function in Eq. (2) is hard to use gradient optimization because the form of recursion [30]. To solve this optimization problem, we turn to Particle Swarm Optimization (PSO) which is inspired by social behavior patterns of organisms that live and interact within large groups. In particular, it incorporates swarming behaviors observed in flocks of birds, schools of fish, or swarms of bees, and even human social behavior, from which the swarm intelligence paradigm has emerged [31, 32]. It could be implemented and applied easily to solve various

Algorithm 2 Sequence recognition under PWHMM

Input: A sequence of tokens O.
Require: Transition probability matrix A, Observation symbols emission distribution B, Initial states distribution Π, Minor Models Vector $MModels$, Likelihood Ratio LRs
Output: Best state sequence solution Q_{best}.

```
01. Q_0=Weighted-Viterbi(O, 1) //get tags produced by HMM
02. Score_0=MModels.score(Q_0), best_Fitness = −∞
03. Search R in Ratio Interval //represent R by PSO's particles
04. foreach Fitness Evaluation do
05.     (Q, L)=Weighted-Viterbi(O, R)
06.     for i in length(MModels)
07.         if MModels[i].score(Q) > Score_0[i]
08.             logLR[i] = logLR[i] + (log(LRs[i][1]) − log(LRs[i][0]))/2
09.         else
10.             logLR[i] = logLR[i] − (log(LRs[i][1]) − log(LRs[i][0]))/2
11.     Fitness = L + Σ_{i=1}^{n} (logLR[i])
12.     if Fitness > best_Fitness:
13.         best_Fitness = Fitness, Q_best = Q
14.     Update positions of particles
15. end foreach
16. Return Q_best
```

function optimization problems, or the problems [33] that can be transformed to function optimization problems. As an algorithm, its main strength is its fast convergence [34, 35], which compares favorably with many other global optimization algorithms. The classical particle swarm model consists of a swarm of particles which are initialized with a population of random candidate solutions. They move iteratively through the d-dimension problem space to search the new solutions, where the fitness f can be calculated as the certain qualities measure. Each particle has a position represented by a position-vector $\vec{x}_i$ (i is the index of the particle), and a velocity represented by a velocity-vector $\vec{v}_i$. Each particle remembers its own best position so far in a vector $\vec{x}_i^{\#}$, and its j-th dimensional value is $x_{ij}^{\#}$. The best position-vector among the swarm so far is then stored in a vector $\vec{x}^*$, and its j-th dimensional value is x_j^*. During the iteration time t, the update of the velocity from the previous velocity to the new velocity is determined by Eq. (6). The new position is then determined by the sum of the previous position and the new velocity by Eq. (7). The pseudo-code for particle swarm optimization algorithm is illustrated in Algorithm 3.

$$\begin{aligned} v_{ij}(t+1) = w v_{ij}(t) + c_1 r_1 (x_{ij}^{\#}(t) - x_{ij}(t)) \\ + c_2 r_2 (x_j^*(t) - x_{ij}(t)) \end{aligned} \tag{6}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{7}$$

Algorithm 3 Particle Swarm Algorithm

01. Initialize the size of the particle swarm n, and other
02. parameters; Initialize the positions and the velocities
03. for all the particles randomly.
04. While (the end criterion is not met) do
05. $t = t + 1$;
06. Calculate the fitness value of each particle;
07. $\vec{x}^* = argmin_{i=1}^{n}(f(\vec{x}^*(t-1)), f(\vec{x}_1(t)),$
08. $f(\vec{x}_2(t)), \cdots, f(\vec{x}_i(t)), \cdots, f(\vec{x}_n(t)));$
09. For i= 1 to n
10. $\vec{x}_i^{\#}(t) = argmin_{i=1}^{n}(f(\vec{x}_i^{\#}(t-1)), f(\vec{x}_i(t));$
11. For $j = 1$ to d
12. Update the j-th dimension value of $\vec{x}_i$ and $\vec{v}_i$
13. according to Eqs.(6),(7)
14. Next j
15. Next i
16. End While.

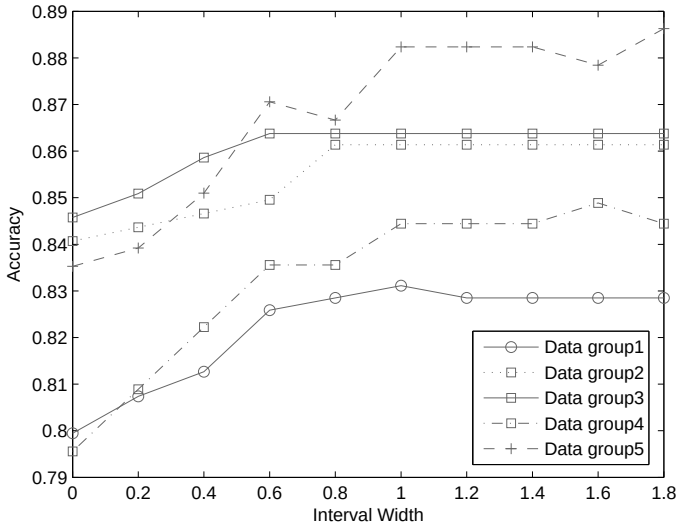

Fig. 2 *The width of Ratio Interval is determined experimentally. A typical case of five data groups is shown.*

	news	reviews	science_fiction	adventure	overall
a	136	142	104	86	468
b	1	4	1	1	7
c	18	11	6	7	42
d	123	104	56	74	357
sensitivity	99.3%	97.3%	99.0%	98.9%	98.5%
specificity	87.3%	90.4%	90.3%	91.4%	89.5%
log(LR(+))	2.05	2.32	2.33	2.44	2.24
log(LR(-))	-4.78	-3.50	-4.55	-4.38	-4.11

Tab. II *Likelihood Ratio of Unknown-pairs Heuristic model based on four categories of dataset.*

categories	description	lexical diversity
news	Chicago Tribune: Society Reportage	4.97
reviews	Time Magazine: Reviews	4.30
science_fiction	Heinlein: Stranger in a Strange Land	4.67
adventure	Field: Rattlesnake Ridge	5.61
mystery	Hitchens: Footsteps in the Night	5.69
romance	Callaghan: A Passion in Rome	4.85
humor	Thurber: The Future, If Any, of Comedy	4.67

Tab. III *The features of the datasets.*

4. Experiment and Discussion

4.1 Datasets

In this experiment, 7 article categories (including news, reviews, etc.) in Brown Corpus are used as datasets. The features of the datasets are shown in Tab. III. This corpus has POS tagged as 70 states. In each article category, the first 200 sentences are used as test set in 10 groups and the following 700 sentences are employed as training set.

4.2 Experimental settings

In targeted POS tagging problem, we choose the NLTK [25] implementation of HMM as baseline. The Brown corpus can be imported into nltk. To keep comparison clear, both the developed method and the baseline use the same supervised training algorithm implemented in NLTK, which is implemented in hmm module.

We use two-stage method. In the first stage, the PWHMMs use weighted ratio to create a set of HMM models. The interval width is determined experimentally as 0.8. As in Fig. 2, five data groups are used to show the typical case of the change of accuracy according to the interval width. The performance starts to become stable with width 0.8 and rarely changes after width 1.2. The experiment shows that the performance of HMM can be improved with Interval Width.

Parameter name	Parameter value
inertia	0.5
cognitive rate	1.0
social rate	1.0
population size	10
max evaluations	50

Tab. IV *Parameter settings for the algorithms.*

groups	news	reviews	science_fiction	adventure	mystery	romance	humor
1	0.8664	0.7787	0.7954	0.8219	0.8234	0.7982	0.7801
2	0.8267	0.7848	0.8179	0.8591	0.8202	0.8606	0.7462
3	0.8720	0.7114	0.7933	0.8618	0.8529	0.7694	0.7348
4	0.8320	0.7419	0.8321	0.8367	0.8512	0.8068	0.7958
5	0.7649	0.8214	0.8533	0.8824	0.8769	0.8204	0.8373
6	0.8377	0.8118	0.7533	0.8849	0.7065	0.7936	0.7958
7	0.8210	0.8431	0.6586	0.8933	0.8683	0.7787	0.8028
8	0.8058	0.7908	0.8191	0.8806	0.9067	0.8604	0.7812
9	0.7546	0.8391	0.7739	0.8643	0.8462	0.8774	0.7772
10	0.8505	0.8263	0.8624	0.8947	0.8880	0.8087	0.7647
Mean	0.8232	0.7949	0.7959	0.8680	0.8440	0.8174	0.7816
Baseline	0.8047	0.7736	0.7712	0.8425	0.8251	0.7963	0.7625

Tab. V *Accuracy details.*

In the second stage, the likelihood ratio LRs is calculated to be [2.24, −4.11] using four categories of corpus in Tab. II. We use unknown pairs heuristic as the only simple minor model. To make the fitness function evaluate the solution of sequence recognition by minor models as well as by HMM, the likelihood ratio test is designed to get the weights of the likelihood modification. In the calculation of likelihood ratio, the feature is positive when the number of unknown pairs of the candidate path is larger than the number of the original path, and is negative when less. The LRs is calculated using a, b, c and d according to Tab. I and Eq. (3). Afterward PSO parameters are set as in Tab. IV. The PSO convergence is always fast due to the dimension in this problem. When we choose population size as 10, the maximum evaluation 50 is enough to converge.

4.3 Results

The performance metric used in this study is the accuracy of the prediction of token-tag pairs. Seven categories in the corpus are used to show that the proposed method is robust in improving the performance of HMM. The results regarding the performance of the developed method are reported in Tab. V. Fig. 3 compares the mean accuracies of the developed method with that of HMM. As illustrated, the proposed method always performs better than HMM.

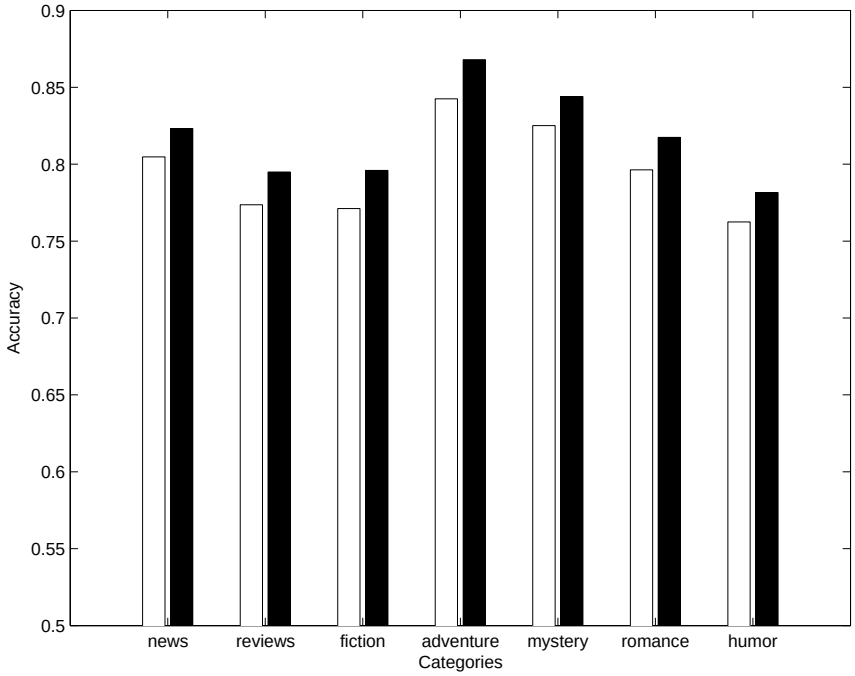

Fig. 3 *Mean accuracy.*

To show the stability of the developed method, Figs. 4, 5, 6 and 7 report the four runs of whole test sets on four corpus categories. As in Tab. VI, the standard deviation is very small. Although random search is used in the second stage, the performance of each run changes very little and always stands above the baseline.

4.4 Discussion

Our proposed method clearly outperforms the baseline on various text categories. This improvement comes from the usage of knowledge in our approach. The reason of stability is that our approach avoids heavy usage of intervals. It exploits the structure of HMM to keep the adjustment simple, i.e. directly using the ratio between transition probabilities and emission probabilities.

By presenting the PWHMM and searching for best ratio weighting, the performance of HMM is improved with the help of a minor model. The validity of the two-stage model selection with HMM and likelihood ratio is confirmed.

5. Conclusion

The proposed approach is the re-optimization of classical HMM. Compared with HMM, our approach has two advantages. First, our approach is more robust to the influence of sparseness and noisiness of training data. Since the learned parameters are extended into intervals, better solutions can be provided by a set of HMM

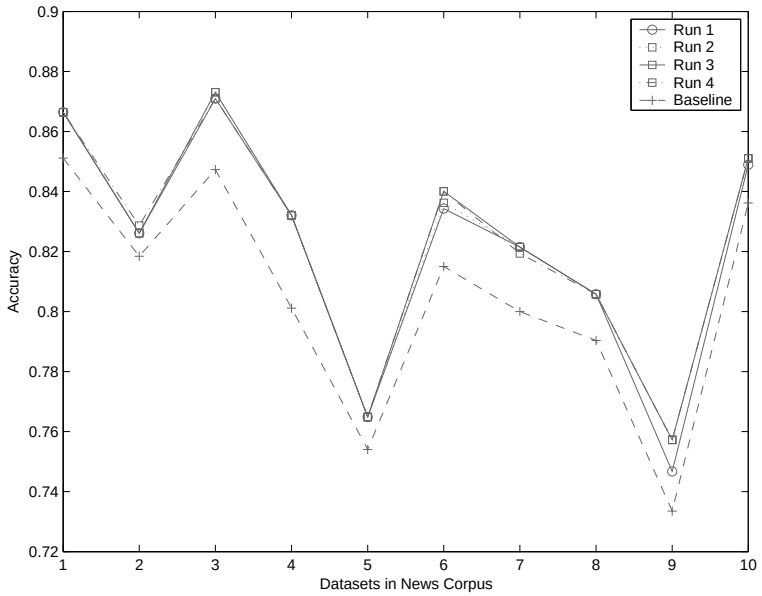

Fig. 4 *Comparison of four runs of the two-stage method and HMM baseline on four categories of corpus.*

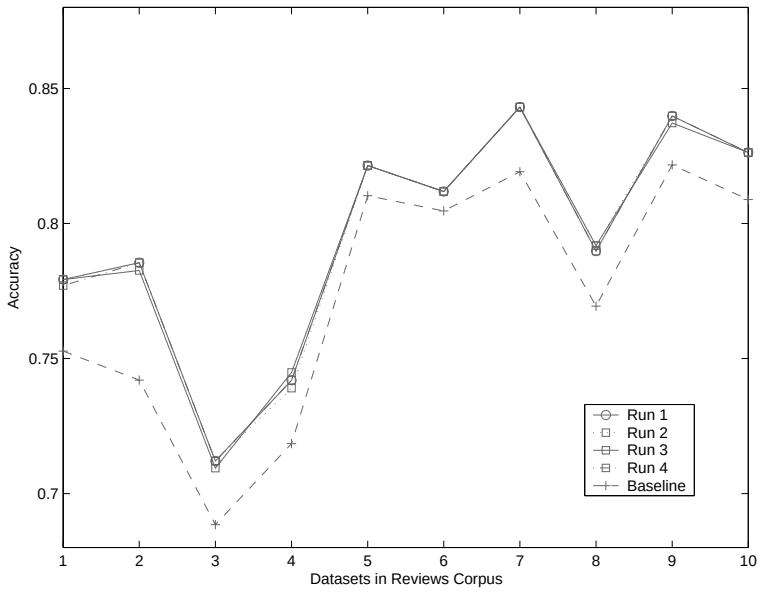

Fig. 5 *Comparison of four runs of the two-stage method and HMM baseline on four categories of corpus.*

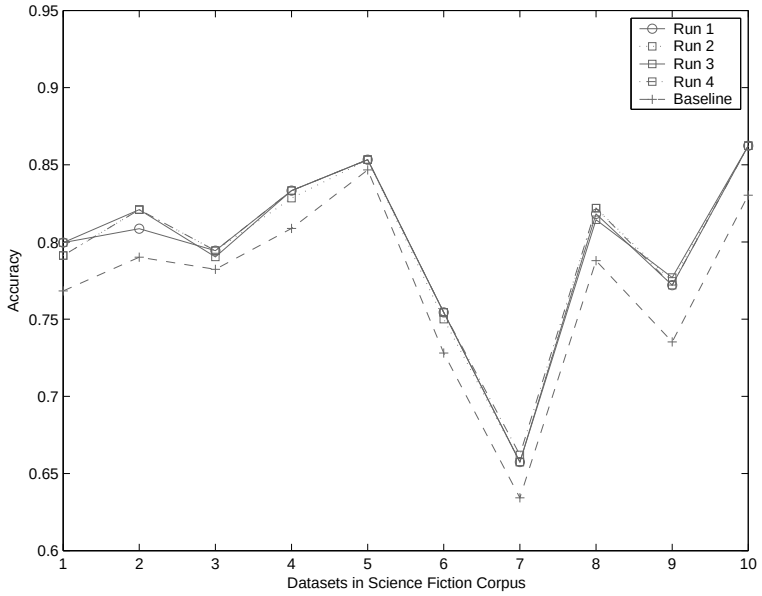

Fig. 6 *Comparison of four runs of the two-stage method and HMM baseline on four categories of corpus.*

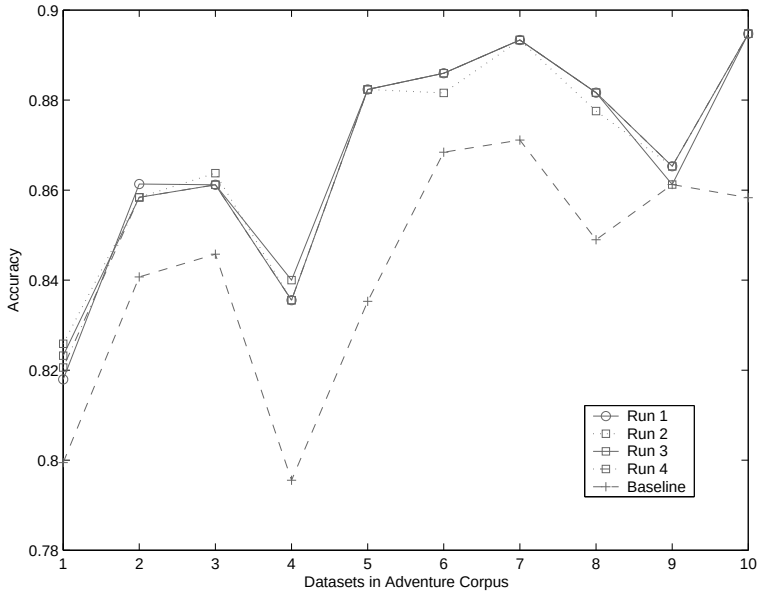

Fig. 7 *Comparison of four runs of the two-stage method and HMM baseline on four categories of corpus.*

groups	news	reviews	fiction	adventure
1	0.0000	0.0011	0.0048	0.0034
2	0.0013	0.0014	0.0062	0.0015
3	0.0012	0.0013	0.0020	0.0013
4	0.0000	0.0024	0.0025	0.0022
5	0.0000	0.0000	0.0000	0.0000
6	0.0029	0.0000	0.0022	0.0022
7	0.0011	0.0000	0.0023	0.0000
8	0.0000	0.0012	0.0036	0.0020
9	0.0053	0.0013	0.0023	0.0020
10	0.0011	0.0000	0.0000	0.0000
Mean	0.0013	0.0009	0.0026	0.0015

Tab. VI *Stability details. Standard deviation of 4 runs over four categories of corpus in 10 data groups.*

models, i.e. PWHMM. Second, it is easier to use features and rules in our approach than in HMM. Features and rules can be adapted into our approach as minor models.

According to the experiment, our two-stage model selection method robustly outperforms the baseline on various text categories. The performance of the proposed method is stable among each runs of random search. Besides, the output state sequence Q is based on the *single* best state sequence criterion and keeps context information not impaired when using minor models.

Acknowledgment

This work is supported partly by the National Natural Science Foundation of China (Grant No. 61173035, No. 60973068), the Fundamental Research Funds for the Central Universities (Grant No. 2012TD027), the Program for New Century Excellent Talents in University, and Dalian Science and Technology Fund (Grant No. 2010J21DW006).

References

[1] Kim J., Rim H., Tsujii J.: Self-organizing Markov models and their application to part-of-speech tagging. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics, Volume 1, Association for Computational Linguistics, 2003, pp. 296–302.

[2] Zhou G., Su J.: Error-driven hmm-based chunk tagger with context-dependent lexicon. In: Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, 2000, pp. 71–79.

[3] Freitag D., McCallum A.: Information extraction with HMM structures learned by stochastic optimization. In: Proceedings of the National Conference on Artificial Intelligence, Menlo Park, CA, 2000, pp. 584–589.

[4] Couvreur L., Couvreur C.: Blind model selection for automatic speech recognition in reverberant environments. Journal of VLSI Signal Processing Systems for Signal, Image, and Video Technology, **36**, 2004, pp. 189–203.

[5] Altman R.: Assessing the goodness-of-fit of hidden Markov models. BioMetrics, **60**, 2004, pp. 444–450.

[6] Baillie M., Jose J., van Rijsbergen C.: HMM model selection issues for soccer video. In Enser, P. and Kompatsiaris, Y. and OConnor, N.E. and Smeaton, A.F. and Smeulders, A.W.M., ed.: Image and Video Retrieval, Proceedings. Volume 3115 of Lecture Notes in Computer Science. 2004, pp. 70–78.

[7] Buza K., Nanopoulos A., Horváth T., Schmidt-Thieme L.: GRAMOFON: General model-selection framework based on networks. Neurocomputing, **75**, 2012, pp. 163–170.

[8] Chien J., Furui S.: Predictive hidden Markov model selection for speech recognition. IEEE Transactions on Speech and Audio Processing, **13**, 2005, pp. 377–387.

[9] Wang S., Zhao Y.: Online Bayesian tree-structured transformation of HMMs with optimal model selection for speaker adaptation. IEEE Transactions on Speech and Audio Processing, **9**, 2001, pp. 663–677.

[10] Ueberla J. P.: Domain adaptation with clustered language models. In: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, 1997, pp. 807–810.

[11] Jair Escalante Hugo, Montes Manuel E. S. L.: Particle swarm model selection. Journal of Machine Learning Research, 2009, pp. 405–440.

[12] Freitag D., McCallum A.: Information extraction with hmms and shrinkage. In: Proceedings of the AAAI-99 Workshop on Machine Learning for Information Extraction, Orlando, Florida, 1999, pp. 31–36.

[13] Li J., Wang J., Zhao Y., Yang Z.: Self-adaptive design of hidden markov models. Pattern Recognition Letters, **25**, 2004, pp. 197–210.

[14] Zhu H., Wang J., Yang Z., Song Y.: A method to design standard hmms with desired length distribution for biological sequence analysis. Algorithms in Bioinformatics, WABI2006, LNCS, 2006, pp. 24–31.

[15] Brants T.: Tnt: a statistical part-of-speech tagger. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, Association for Computational Linguistics, 2000, pp. 224–231.

[16] Chang M., Goldwasser D., Roth D., Srikumar V.: Structured output learning with indirect supervision. In: Proceedings of the International Conference on Machine Learning, IEEE Computer Society, 2010, pp. 199–206.

[17] Salvetti F., Lewis S., Reichenbach C.: Impact of lexical filtering on overall opinion polarity identification. In: Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications, Stanford, US., 2004.

[18] Lin H., Zhan X., Yao T.: Example-based Chinese text filtering model. In: Proceedings of the 5th International Computer Science Conference, Springer, 1999, pp. 415–420.

[19] McCallum A., Freitag D., Pereira F.: Maximum entropy Markov models for information extraction and segmentation. In: Proceedings of the Seventeenth International Conference on Machine Learning, IEEE, 2000, pp. 591–598.

[20] Lafferty J., McCallum A., Pereira F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: Proceedings of the Eighteenth International Conference on Machine Learning, IEEE, 2001, pp. 282–289.

[21] Rabiner L.: A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, **77**, 1989, pp. 257–286.

[22] Juang B., Rabiner L.: Fundamentals of speech recognition. Prentice Hall, Englewood Cliffs, NJ, 1993.

[23] Baum L., Petrie T.: Statistical inference for probabilistic functions of finite state markov chains. The Annals of Mathematical Statistics, **37**, 1966, pp. 1554–1563.

[24] Baum L., Eagon J.: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology. Bulletin of American Mathematical Society, **73**, 1967, pp. 360–363.

[25] Loper E., Bird S.: NLTK: The Natural Language Toolkit. In: Proceedings of the ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, Somerset, NJ: Association for Computational Linguistics, 2002, pp. 62–69.

[26] Cheng C., Li S.: Fuzzy Time Series Forecasting With a Probabilistic Smoothing Hidden Markov Model. IEEE T. Fuzzy System, **20**, 2012, pp. 291–304.

[27] Rafiul Hassan M., Nath B., Kirley M., Kamruzzaman J.: A hybrid of multiobjective Evolutionary Algorithm and HMM-Fuzzy model for time series prediction. Neurocomputing, **81**, 2012, pp. 1–11.

[28] Zeng J., Liu Z.: Type-2 fuzzy hidden Markov models and their application to speech recognition. IEEE Transactions on Fuzzy Systems, **14**, 2006, pp. 454–467.

[29] Li W., Liu L., W., G.: Multi-objective uniform design as a SVM model selection tool for face recognition. Expert Systems with Applications, **38**, 2011, pp. 6689–6695.

[30] Wozniak M., Zmyslony M.: Combining classifiers using trained fuser – analytical and experimental results. Neural Network World, **20**, 2010, pp. 925–934.

[31] Fogel D.: Evolutionary Computation: Toward a New Philosophy of Machine Intelligence. Volume 1. J. Wiley, IEEE Press, 2006.

[32] Clerc M.: Particle Swarm Optimization. ISTE Publishing Company, London, 2006.

[33] Chen M. Y.: A hybrid model for business failure prediction – utilization of particle swarm optimization and support vector machines. Neural Network World, **21**, 2011, pp. 129–152.

[34] Yue B., Liu H., Abraham A., Badr Y.: A multi-swarm synergetic optimizer for multi-knowledge extraction using rough set. Neural Network World, **20**, 2010, pp. 501–517.

[35] Sun S., Abraham A., Zhang G., Liu H.: A particle swarm optimization algorithm for neighbor selection in peer-to-peer networks. In: Proceedings of the Sixth Computer Information Systems and Industrial Management Applications, IEEE Computer Society, 2007, pp. 166–172.