Improving Handwritten Chinese Text Recognition by Unsupervised Language Model Adaptation

Qiu-Feng Wang, Fei Yin, Cheng-Lin Liu
National Laboratory of Pattern Recognition (NLPR)
Institute of Automation of Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
{*wangqf, fyin, liucl*}*@nlpr.ia.ac.cn*

***Abstract*—This paper investigates the effects of unsupervised language model adaptation (LMA) in handwritten Chinese text recognition. For no prior information of recognition text is available, we use a two-pass recognition strategy. In the first pass, the generic language model (LM) is used to get a preliminary result, which is used to choose the best matched LMs from a set of pre-defined domains, then the matched LMs are used in the second pass recognition. Each LM is compressed to a moderate size via the entropy-based pruning, tree-structure formatting and fewer-byte quantization. We evaluated the LMA for five LM types, including both character-level and word-level ones. Experiments on the CASIA-HWDB database show that language model adaptation improves the performance for each LM type in all domains. The documents of ancient domain gained the biggest improvement of character-level correct rate of 5.87 percent up and accurate rate of 6.05 percent up.**

***Keywords*-Handwritten Chinese text recognition; Two-pass recognition; Language model adaptation; Language model compression**

I. Introduction

Handwritten Chinese character recognition has attracted much attention from the 1970s and has achieved tremendous advances [1], [2]. However, most works were on the Chinese isolated character recognition, the works on Chinese character string recognition were mostly aimed for the recognition in rather constrained application domain, such as legal amount recognition in bank checks [3] and address phrase recognition [4], both are with very strong lexical constrains. In Chinese handwritten recognition of general texts, the works have been reported only in recent years, and the reported accuracies are quite low (e.g. character-level correct rate of 39.37% in [5]). Although our recent work by integrating multiple contexts (e.g., language model) achieved a much higher correct rate of 91.17% [6], there are still many recognition errors due to the mismatch between the language model and recognition text domain. To manage this mismatch problem, language model adaptation in handwritten text recognition is investigated in this paper. To our best of knowledge, there is no work about language model adaptation in handwritten text recognition.

Many studies have been conducted for language model adaptation in speech recognition [7] and natural language processing [8], [9]. One method is supervised language model adaptation, where topic information is typically available and a topic specific language model is interpolated with the generic one (e.g., [8]). In contrast, various unsupervised approaches perform latent topic analysis for language model adaptation (e.g., [10]), or use the transcription result directly to estimate an adapted language model [11] or lexicon [12]. Unsupervised adaptation is more relevant for real applications where the topic is unknown a priori. For only limited in-domain data (domain matched with the recognition task) is usually available for language model adaptation [7], the language modeling community is showing a growing interest in collecting text from Internet to supplement sparse in-domain resources [13]. In Chinese, Sogou Labs[1] provides a large set of resources about diverse domains extracted from the Internet.

This paper reports our first attempt to unsupervised language model adaptation for handwritten Chinese text recognition (HCTR) to improve the recognition performance, particularly for those texts with different context from common language. We consider a two-pass recognition strategy for this adaptation. The first-pass recognition result by the generic language model (LM) is used to choose the best matched LM from a set of pre-defined language models. These language models are estimated on a large text resource with pre-defined diverse domains from Sogou Labs, and each language model is compressed by three steps: the entropy-based pruning [14], tree-structure formatting and fewer-byte quantization [15]. In our experiments on the CASIA-HWDB database, both character-level and word-level LMs are considered. The results show that the unsupervised adaptation for these language models benefits the text recognition performance, especially for the ancient domain text recognition, which is very mismatched with the generic language model.

II. System Overview

This work is based on our previous system [6], and the block diagram is shown in Fig. 1. For the unsupervised language model adaptation, the two-pass strategy is used to recognize each input document. In the first pass, a generic language model is used to get a preliminary recognition

[1] http://www.sogou.com/labs/resources.html

978-0-7695-4661-2/12 $26.00 © 2012 IEEE
DOI 10.1109/DAS.2012.46

result, then we choose language models best matched with that result, which are used in the second pass.

In the first pass recognition, we take the following seven steps (only the last three steps are needed for the second pass recognition): (1) each text line image is extracted; (2) the line image is over-segmented into a sequence of primitive segments (Fig. 2a); (3) consecutive segments are combined to generate candidate character patterns (Fig. 2b); (4) each candidate pattern is classified to several candidate character classes, forming a character candidate lattice (Fig. 2c); (5) for word-level language model, each sequence of candidate characters is matched with a lexicon to segment into candidate words, forming a word candidate lattice (Fig. 2d); (6) each character sequence or word sequence **C** paired with candidate pattern sequence **X** (the pair is called a candidate segmentation-recognition path) is evaluated by multiple contexts, and the optimal path is searched to give the segmentation and recognition result; (7) all text lines results are concatenated to give the document result, which is used for language model adaptation or output.

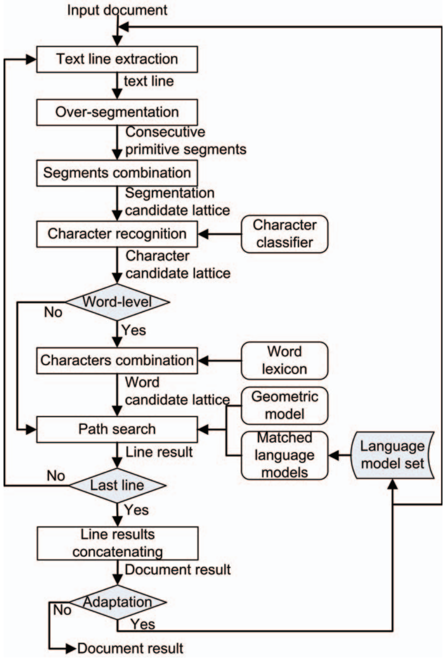
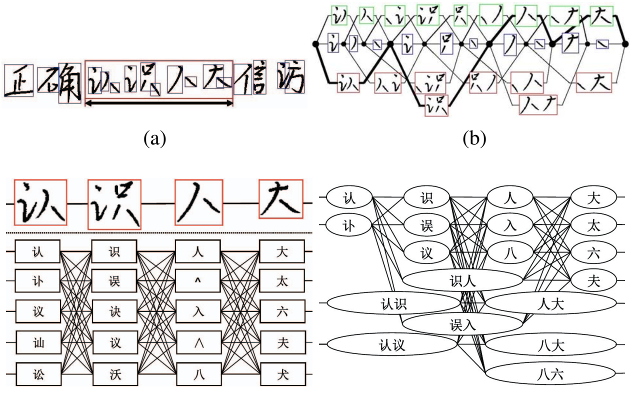


Figure 1: System diagram for handwritten Chinese text recognition.

In this work, we evaluate each segmentation-recognition path by integrating character recognition score, geometric context and linguistic context [6]:

$$f(X^s, C) = \sum_{i=1}^{m} (w_i \cdot lp_i^0 + \sum_{j=1}^{4} \lambda_j \cdot lp_i^j) + \lambda_5 \cdot \log P(C), \quad (1)$$

where w_i is the width of the i-th character pattern after normalizing by the estimated height of the text line, and $lp_i^0 = \log p(c_i|\mathbf{x}_i)$ is character recognition score, $lp_i^1 = \log p(c_i|g_i^{\text{uc}})$, $lp_i^2 = \log p(c_{i-1}c_i|g_i^{\text{bc}})$, $lp_i^3 = \log p(z_i^{\text{p}} = 1|g_i^{\text{ui}})$, and $lp_i^4 = \log p(z_i^{\text{g}} = 1|g_i^{\text{bi}})$ are four geometric model scores, and $\log P(C)$ denotes the language model score. The combining weights $\lambda_j, j = 1, \ldots, 5$ are optimized by Maximum Character Accuracy training [6].



(a) (b)

(c) (d)

Figure 2: (a) Over-segmentation; (b) Segmentation candidate lattice; (c) Character candidate lattice of a segmentation (thick path) in (b); (d) Word candidate lattice of (c).

Under this path evaluation function, we use a refined beam search method [6] to find the optimal path. The search proceeds in frame-synchronous fashion with two steps pruning: first, we only keep the best partial path at each candidate character, then keep the top beam-width partial paths at each segmented point.

III. Adaptation Approach

In this paper, we use the n-gram language model, which has been successful used in our previous works [6], [16], and five types of n-grams are evaluated for adaptation: character bi-gram (**cbi**) and tri-gram (**cti**), word bi-gram (**wbi**) and word class bi-gram (**wcb**), interpolating word and class bi-gram (**iwc**). The details are shown in Table I, where $C =< c_1 \ldots c_m >$ is a character sequence, and m is the character number of C (In word level, C is segmented to a word sequence $C =< w_1 \ldots w_l >$, and l is the word number, and W_i is the word class of the word w_i). To match the domain of recognition text, the language model is desired to be changed dynamically for different texts. Under the assumption that the domain in one recognition document is the same, we use a two-pass recognition strategy in each document for unsupervised adaptation of language model.

A. Language Model Adaptation

In the two-pass recognition strategy, we can get the automatic transcript of each document after the first pass. Although this transcript is a very direct in-domain data, there are too few characters in each document (usually 200-300 characters) to obtain an appropriate n-gram. However, we can use this transcript to choose the best matched language

Table I: The n-grams used in our experiments.

level	n-gram	formula
character	**cbi**	$P_{cbi}(C) = \prod_{i=1}^{m} p(c_i \vert c_{i-1})$
	cti	$P_{cti}(C) = \prod_{i=1}^{m} p(c_i \vert c_{i-2}c_{i-1})$
word	**wbi**	$P_{wbi}(C) = \prod_{i=1}^{l} p(w_i \vert w_{i-1})$
	wcb	$P_{wcb}(C) = \prod_{i=1}^{l} p(w_i \vert W_i)p(W_i \vert W_{i-1})$
	iwc	$\log P_{iwc}(C) = \log P_{wbi}(C) + \lambda_6 \cdot \log P_{wcb}(C)$

model from a pre-defined set ($LM_k, k = 1, ..., K, K$ is the class number of pre-defined domains, and LM_0 is for the generic one). The details of this unsupervised language model adaptation (LMA) method is described as follows.

Two-Pass recognition for unsupervised LMA

1) Use the generic LM (LM_0) to recognize a document to obtain a preliminary transcript.
2) Use the transcript to choose the best matched language model (LM_{k^*}) to maximize the log-likelihood (2).
3) Use LM_{k^*} to recognize the document again to obtain the final transcript.

Here we assume the preliminary transcript after the first pass recognition is C, then the best matched LM can be chosen by maximizing the log-likelihood:

$$k^* = \arg\max_k \log P_k(C),\ 0 \le k \le K \tag{2}$$

where $P_k(C)$ is the k-th language model probability of transcript C. This criterion is the same as choosing the language model of the minimum perplexity.[2]

Sometimes only one LM is not enough, we choose the best two LMs ($LM_{ki}, i = 1, 2$) according to (2) , and such two LMs are used by linear interpolation in the second-pass recognition:

$$P(C_2) = \lambda \cdot P_{k1}(C_2) + (1 - \lambda) \cdot P_{k2}(C_2), \tag{3}$$

where C_2 is a candidate transcription in the second pass recognition, and the weight λ is used to balance such two LMs, and is calculated by the perplexity:

$$\lambda = \frac{PP_{k2}(C)}{PP_{k1}(C) + PP_{k2}(C)}, \tag{4}$$

where the function $PP_{ki}(C)$ denotes the perplexity of LM_{ki} on the first pass result sequence C:

$$PP_k(C) = P_k(C)^{-\frac{1}{m}} = \sqrt[m]{\frac{1}{\prod_{i=1}^{m} p_k(c_i \vert c_1^{i-1})}}. \tag{5}$$

B. Language Model Compression

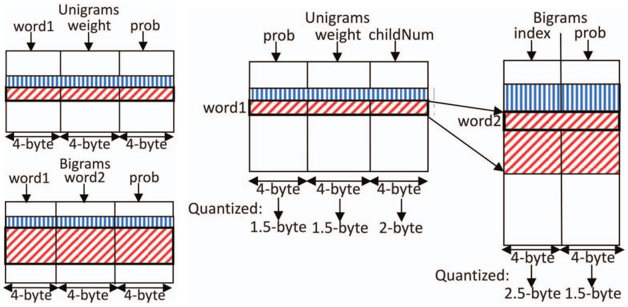
Due to the large storage of all $K+1$ LMs, we compress each LM with three steps: the entropy-based pruning [14], tree-structure formatting and fewer-byte quantization [15].

The entropy-based pruning removes those n-grams rasing the perplexity (due to prune them) less than a threshold. In our previous work [16], it is demonstrated that an appropriate threshold can yield good tradeoff between the model size and the performance. Using the SRILM toolkit [17], we can easily compress per n-gram with the entropy-based pruning.

However, the output of SRILM is the list-structure, where many prefixes are repeated in the bi-gram (Fig. 3a). To remove such duplicate space, we format each n-gram as the tree-structure (Fig. 3b). The tree-structure originates from a hypothetical root node (not shown) which branches out into the uni-gram nodes at the first level, each of which branches out to bi-gram nodes at the second level and so on.

In the tree-structure, each element of per node generally uses 4-byte representation in the 32-bit architecture. To further save the storage, we use the fewer-byte representation to store each element (e.g., word bi-gram in Fig. 3b). In character bi-gram, the 'index' uses 13-bit enough to represent all character classes (7,356) in our experiment, and 'prob' uses 11-bit to sum up to a 3-byte representation.

Finally, the storage sizes of all 16 LMs (see Section IV) for each type are shown in Table II, where the last column shows the storage ratio for each LM type after compression (**wcb** is without the entropy-based pruning due to its moderate model size [16]). After compression, the average sizes of per n-gram are 2.1MB, 7.2MB, 4.4MB, 1.9MB and 6.3MB for **cbi**, **cti**, **wbi**, **wcb** and **iwc**, respectively.



(a) (b)

Figure 3: word bi-gram storage structures: (a) List-structure; (b) Tree-structure and quantization.

Table II: The storage size (MB) for all 16 n-grams of four types after each step of compression.

	original	pruning	formatting	quantization	ratio
cbi	285	175	89.2	33.3	11.68%
cti	1847	595	310	115	6.23%
wbi	1338	334	183	71.0	5.31%
wcb	129	129	81.2	31.1	24.11%

[2] Perplexity is the most commonly used method to evaluate the performance of a language model, smaller perplexity denotes higher performance.

IV. Experiments

We use the system introduced detailedly in [6] as the baseline (except candidate character augmentation) to evaluate the unsupervised language model adaptation (LMA), and all the experiments are implemented on a desktop computer of 2.66GHz CPU, programming using Microsoft Visual C++.

A. Database and Experimental Setting

We evaluate the performance on a large database CASIA-HWDB [18], which is divided into a training set of 816 writers and a test set of other 204 writers. The training set contains 3,118,477 isolated character samples of 7,356 classes and 4,076 pages of handwritten texts (including 1,080,017 character samples). We tested on the unconstrained texts including 1,015 pages, which were segmented into 10,449 text lines and there are 268,629 characters.

The character classifier used in our system is a modified quadratic discriminant function (MQDF), and the parameters were learned from 4/5 samples of training set, and the remaining 1/5 samples were for confidence parameter estimation. For parameter estimation of the geometric models, we extracted geometric features from all training text pages. The generic LMs were trained on a large corpus from the Chinese Linguistic Data Consortium. On obtaining the context models, the combining weights were learned on 300 pages of training text.

To prepare a pre-defined set of LMs to match different recognition pages, we extracted 14 corpora about different domains from the web pages provided by Sogou Labs. All texts were segmented into the word sequences by ICTCLAS[3] toolkit for word-level LMs, and further, we clustered such words into a number of word classes by the algorithm in [19]. In addition, an ancient domain corpus (without word segmentation due to no ancient domain word table) was collected from the Internet. Finally, Table III shows the statistics of characters, words, character classes, word classes and word clusters in each corpus. We can see that the corpus of news domain is the largest, which has about 418 million characters and 265 million words, and it is much larger than the generic one. On the other hand, the texts of ancient domain are much fewer, however, about 8.22 million characters are enough to get an appropriate character bi-gram and tri-gram using the SRILM [17] toolkit.

We evaluate the recognition performance using two character-level accuracy metrics as in the baseline system [6]: Correct Rate (**CR**) and Accurate Rate (**AR**):

$$\begin{aligned} CR &= (N_t - D_e - S_e)/N_t, \\ AR &= (N_t - D_e - S_e - I_e)/N_t, \end{aligned} \tag{6}$$

where N_t is the total number of characters in the transcript. The numbers of substitution errors (S_e), deletion errors (D_e) and insertion errors (I_e) are calculated by the aligning the recognition result string with the transcript by dynamic programming.

Table III: Statistics of characters, words, character classes, word classes and word clusters in each corpus.

domains	LMs	characters (million)	words (million)	character classes	word classes	word clusters
generic	LM_0	50.8	32.7	7356	281,680	1000
news	LM_1	418	265	6699	454,370	1000
business	LM_2	333	202	6474	473,792	1000
sport	LM_3	227	149	6789	234,130	1000
house	LM_4	118	73.4	6231	254,659	1000
entertain	LM_5	106	71.5	5882	144,246	500
it	LM_6	54.1	33.1	5628	156,728	500
Olympic	LM_7	52.2	33.0	6048	144,390	500
women	LM_8	44.0	29.5	5569	94,409	350
auto	LM_9	32.4	20.4	5153	105,487	500
travel	LM_{10}	31.4	20.1	5755	133,731	500
health	LM_{11}	31.2	20.2	5590	99,207	350
learning	LM_{12}	28.0	17.5	5548	104,282	350
culture	LM_{13}	20.0	13.4	5791	104,162	350
military	LM_{14}	15.3	9.47	4854	69,683	250
ancient	LM_{15}	8.22	—	7318	—	—

B. Experimental Results

We evaluate the effect of the unsupervised LMA approach including both choosing only one LM and two best LMs according to (2), and further, we show the performance improvement of LMA in different domains. We also give the processing time on all test pages (1,015 pages) excluding that of over segmentation and character recognition, which are stored after the first pass recognition.

Table IV shows the results of LMA using only one LM chosen by (2) in the second pass recognition. Compared to the baseline performance (A small difference with [6] is due to the compression of generic language model here), we can see that both CR and AR are improved by the LMA for all LM types, and the improvement for **cbi** is the largest (about 1.0 percent up). On the other hand, the processing time is doubled due to the two-pass recognition strategy.

Table IV: Effects of LMA with only one LM.

	Baseline			LMA-one LM		
LMs	CR(%)	AR(%)	Time(h)	CR(%)	AR(%)	Time(h)
cbi	90.26	89.56	0.27	91.24	90.57	0.54
cti	90.80	90.20	0.36	91.72	91.16	0.76
wbi	90.98	90.33	1.02	91.84	91.23	2.05
wcb	90.80	90.10	1.09	91.68	91.01	2.21
iwc	**91.21**	**90.57**	1.16	92.05	91.44	2.36

The results of LMA using two LMs are shown in Table V. Averagely, about 0.2 percent is improved further, and compared to the baseline system, the best performance of our system (using **iwc**) is improved from 91.21% to 92.19% for CR, and from 90.57% to 91.58% for AR. Again, the largest improvement is got by **cbi** (about 1.2 percent up).

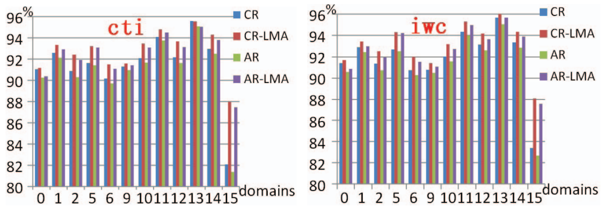
Further, we investigate the effect of the LMA (using two best LMs) for each domain, and the results of **cti** and **iwc** are shown in Fig. 4 (Four domains without any test pages are not shown). We can see that the improvement of ancient

[3] Institute of Computing Technology, Chinese Lexical Analysis System: http://ictclas.org/

Table V: Effects of LMA with the two best LMs.

	LMA-one LMs			LMA-two LMs		
LMs	CR(%)	AR(%)	Time(h)	CR(%)	AR(%)	Time(h)
cbi	91.24	90.57	0.54	91.45	90.78	0.76
cti	91.72	91.16	0.76	91.91	91.34	1.06
wbi	91.84	91.23	2.05	92.02	91.40	2.38
wcb	91.68	91.01	2.21	91.83	91.15	2.64
iwc	92.05	91.44	2.36	**92.19**	**91.58**	2.85

domain (indexed as 15, see Table III) is the largest, this is because the language expression style of these ancient texts are very different from the style of the generic corpus after the long history. Table VI shows the results of LMA for ancient domain pages (For no word-level LMs of ancient domain, we use the **cti** instead of **wbi**, **wcb**, and **iwc** in the second pass recognition), and the largest improvement is gained by **cti**, improving CR and AR by 5.87 and 6.05 percent, respectively.



(a) (b)

Figure 4: Effects of LMA for per domain using (a) cti, (b) iwc.

Table VI: Effects of LMA for ancient domain pages.

	Baseline		LMA-two LMs		Improvement	
LMs	CR(%)	AR(%)	CR(%)	AR(%)	CR(%)	AR(%)
cbi	82.23	81.49	87.73	86.91	5.20	5.42
cti	82.09	81.39	87.96	87.44	**5.87**	**6.05**
wbi	83.09	82.40	88.02	87.52	4.93	5.12
wcb	82.48	81.69	87.95	87.42	5.47	5.73
iwc	83.36	82.66	88.04	87.55	4.68	4.89

V. Conclusion

This paper presented an approach of unsupervised language model adaptation in handwritten Chinese text recognition system using two-pass recognition strategy with a pre-defined set language models. Each language model is compressed to a moderate size after three compression steps. The second pass recognition gives the improved performance due to the better matched language models than the generic one, especially for the ancient domain pages, because their language style is very different from the genetic corpus. Since all language models used in this paper only consider short distance (one or two) history characters or words, based on the language model adaptation, our future work will integrate long distance contextual information to further improve the handwritten text recognition performance.

Acknowledgment

This work was supported by the National Natural Science Foundation of China (NSFC) under Grant 60825301 and Grant 60933010, National Basic Research Program of China (973 Program) Grant 2012CB316302.

References

[1] R.-W. Dai, C.-L. Liu, B.-H. Xiao, Chinese Character Recognition: History, Status and Prospects, *Frontiers of Computer Science in China*, vol.1, no.2, pp.126-136, 2007.

[2] H. Fujisawa, Forty Years of Research in Character and Document Recognition—An Industrial Perspective, *Pattern Recognition*, vol.41, no.8, pp.2435-2446, 2008.

[3] H.-S. Tang, E. Augustin, C.Y. Suen, O. Baret, M. Cheriet, Spiral Recognition Methodology and Its Application for Recognition of Chinese Bank Checks, *Proc. 9th IWFHR*, pp.263-268, Oct, 2004.

[4] C.-H. Wang, Y. Hotta, M. Suwa, S. Naoi, Handwritten Chinese Address Recognition, *Proc. 9th IWFHR*, pp.539-544, Oct, 2004.

[5] T.-H. Su, T.-W. Zhang, D.-J. Guan, H.-J. Huang, Off-Line Recognition of Realistic Chinese Handwriting Using Segmentation-Free Strategy, *Pattern Recognition*, vol.42, no.1, pp.167-182, 2009.

[6] Q.-F. Wang, F. Yin, C.-L. Liu, Handwritten Chinese Text Recognition by Integrating Multiple Contexts, *IEEE Trans. Pattern Anal. Mach. Intell.*, accepted, Nov, 2011.

[7] J.R.Bellegarda, Statistical Language Model Adaptation: Review and Perspectives, *Speech Communication*, vol.42, no.1, pp.93-108, 2004.

[8] J.F. Gao, H.Suzuki, W. Yuan, An Empirical Study on Language Model Adaptation, *ACM Trans. Asian Language Information Processing*, vol.5, No.3, pp.209-227, 2005.

[9] F.-F. Liu, Y. Liu, Unsupervised Language Model Adaptation Incorporating Named Entity Information, *Proc. 45th ACL*, pp.672-679, Jun, 2007.

[10] D. Mrva, P.C. Woodland, Unsupervised Language Model Adaptation for Mandarin Broadcast Conversation Transcription, *Proc. of Interspeech*, pp.2206-2209, 2006.

[11] M. Bacchiani, B. Roark, Unsupervised Language Model Adaptation, *Proc. ICASSP*, pp.224-227, 2003.

[12] P. Xiu, H. Baird, Incorporating Linguistic Model Adaptation into Whole-Book Recognition, *Proc. 20th ICPR*, pp.2057-2060, Aug, 2010.

[13] A.Sethy, P.G. Georgiou, B.Ramabhadran, S.Narayanan, An Iterative Relative Entropy Minimization-Based Data Selection Approach for n-Gram Model Adaptation, *IEEE Trans. Audio, Speech, Language Processing*, vol.17, no.1, 2009.

[14] A. Stolcke, Entropy-Based Pruning of Backoff Language Models, *Proc. DARPA Broadcast News Transcription and Understanding Workshop*, pp.270-274, 1998.

[15] E.W.D. Whittaker, B. Raj, Quantization-based Language Model Compression, *Proc. of Eurospeech*, pp.33-36, 2001.

[16] Q.-F. Wang, F. Yin, C.-L. Liu, Integrating Language Model in Handwritten Chinese Text Recognition, *Proc. 10th ICDAR*, pp.1036-1040, Jul, 2009.

[17] A. Stolcke, SRILM - an extensible language modeling toolkit, *Proc. 7th ICSLP*, pp.901-904, Sep, 2002.

[18] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, CASIA Online and Offline Chinese Handwriting Databases, *Proc. 11th ICDAR*, pp.37-41, Sep, 2011.

[19] S. Martin, J. Liermann, H. Ney, Algorithms for Bigram and Trigram Word Clustering, *Speech Communication*, vol.24, no.1, pp.19-37, 1998.