A Hybrid Language Model for Handwritten Chinese Sentence Recognition

Qizhen He, Shijie Chen, Mingxi Zhao, Wei Lin
Software Solution Team, Motorola Solutions Inc., Shanghai, China
rmpq64@motorolasolutions.com

Abstract

In this paper, we propose a hybrid language model for handwritten Chinese sentence recognition. This hybrid model is integrated from several independent language models, each of which is trained from a distinct type of corpus and models specifically the linguistic behavior for that kind of corpus. By inferring the type of the string which the user has already written, we can make this hybrid language model contribute more precisely to the recognition engine. To improve the recognition accuracy, we also propose a candidate re-ranking process after recognition by reducing the language scores. Our experiments show that the hybrid language model performs consistently well among different types of handwritten articles, and the overall performance is significantly better than a single standard language model. The result also demonstrates that the candidate re-ranking process effectively improves the performance of the recognition engine in terms of accuracy.

1. Introduction

Nowadays people are using more and more touch-screen devices, such as tablet, smart phones, and etc. Those large screens allow users to write a whole sentence instead of just a character or word, improving user experience greatly. As a result, techniques for good handwritten sentence recognition are necessary. Although character recognition, which has been studied for decades, has achieved satisfying recognition rate for commercial products, sentence recognition remains a research problem. The main difficulty is to have sentences segmented into words correctly before they are recognized. Many researchers devoted themselves to solving the problem [1-7]. Among them, two kinds of information are commonly used and combined to get better recognition results: Shape Recognition and Linguistic Knowledge.

For shape recognition, it is difficult to segment a text line into characters because space between characters is not obvious and some characters are connected in cursive writing. In [8], Su. et al introduced a Segmentation-Free method that combined a hidden Markov Model to recognize a text line. This method is to slice and label a text line into frames evenly. The labeled frames are then concatenated into characters during recognition. However, this method does not incorporate the character shape information sufficiently. On the other hand, [9] introduced the Segmentation-Based method by over-segmenting a text line into primitive segments, each of which represents a character or a part of a character and is recognized using the Character-Recognize Engine. Those primitive segments are then combined and evaluated according to both geometric and the linguistic context [10].

To introduce the linguistic knowledge, language model, modeling the language behavior statistically, is commonly adopted [11]. The probability of a term is first translated into a language score which measures the correctness of the term from linguistic aspect, and then combined with the shape recognition score to search for the most probable recognition result.

However, like most statistical models do, the performance of language model is sensitive to the training data. For example, if the language model is trained from News corpus, it would be no use if not harmful when the users want to write a piece of academic paper. Although incorporating corpus from academic paper into training data would help, there is still a problem to balance these two kinds of corpus, e.g. it is not easy to decide the amount of each kind of corpus in the training set.

To solve this problem, the primary contribution of this paper is that, we propose a hybrid language model for handwritten Chinese sentence recognition. In our

978-0-7695-4774-9/12 $26.00 © 2012 IEEE
DOI 10.1109/ICFHR.2012.157

approach, several language models are built from different types of corpus separately. Based on the stings the user has already written, we infer which type of text he is writing, and combine the models to contribute to the language score. The advantage of this approach is that, by doing so, we can make a good balance between different kinds training corpus, and we can make the language model contribute more precisely to the recognition.

Furthermore, we also propose a post-operation process to re-rank the candidates after recognition by reducing the value of language score. The reason is that during the recognition process (i.e., over-segmentation and pruning based dynamic programming) we need a high language score to keep linguistically correct paths from being cut off, while during the post-operation process language score should be reduced to make the recognition result consistent with shape recognition. Our experiments show that this is a good balance between the shape recognition score and the language score.

The rest of this paper is organized as follows. Section 2 and Section 3 present the construction of the hybrid language model and parameter selection respectively. We show our experiment results in Section 4, and conclude our paper in Section 5.

2. A hybrid Language Model

In this section, after a brief introduction on a typical dynamic-programming (DP for short) based sentence recognition system which incorporates a normal language model, we introduce our proposed hybrid language model.

2.1. Language model in DP based sentence recognition

Here we only summarize the basic ideas of the DP based sentence recognition system. For details, we refer to [6].

The input (i.e., the strokes) is segmented into several primitive segments according to some geometric features. Each primitive segment or segment combination is recognized by a character-level engine to get several candidates. Then a candidate lattice is constructed based on the segmentation points and the character candidates (Figure 1). Each path, which represents a candidate string of the sentence, is evaluated from both shape and linguistic aspects and then associated with a score. The path with the highest score is regarded as the recognition result.

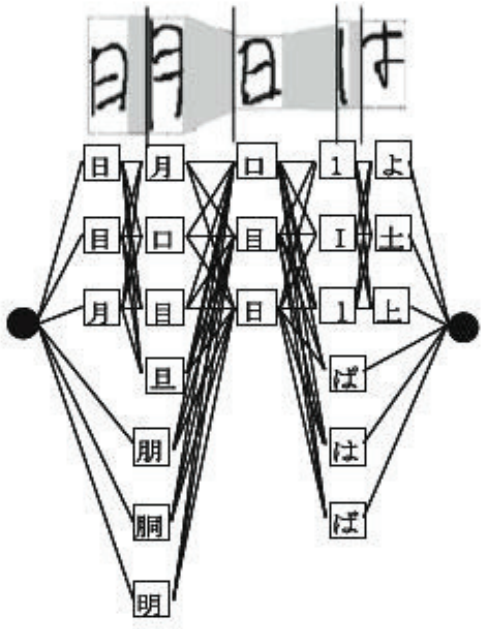

Figure 1. Candidate lattice of DP based recognition

For linguistic evaluation, string probability (provided by language model) is used to measure the grammatical correctness of the string. Given a string s with m characters $s = (w_1 w_2 ... w_m)$, a bi-gram language model assigns to s a probability $P(w_1 w_2 ... w_m)$ as defined in the following equation

$$\begin{aligned} P(w_1 w_2 ... w_m) &= \prod_{i=1}^{m} P(w_i \mid w_1 ... w_{i-1}) \\ &\approx \prod_{i=1}^{m} P(w_i \mid w_{i-1}) \end{aligned} \quad (1)$$

where $P(w_i \mid w_1 ... w_{i-1})$ and $P(w_i \mid w_{i-1})$ are conditional probabilities. The probability $P(w_1 w_2 ... w_m)$ represents the prior probability of the occurrence of s in the training corpus.

Then the score of the path is finally given by:

$$S(w_1 w_2 ... w_m) = S_{shape}(w_1 w_2 ... w_m) + \alpha P(w_1 w_2 ... w_m) \quad (2)$$

where $S_{shape}(w_1 w_2 ... w_m)$ is the score from shape evaluation, and α is the combining weights.

2.2. A Hybrid Language Model

As mentioned before, to make the language model contribute more precisely to the recognition, we propose to build several language models separately from different type of training corpus. And based on our inferring on which type of text the user is writing, we calculate a proper language score. The prototype of the hybrid model can be illustrated in Figure 2.

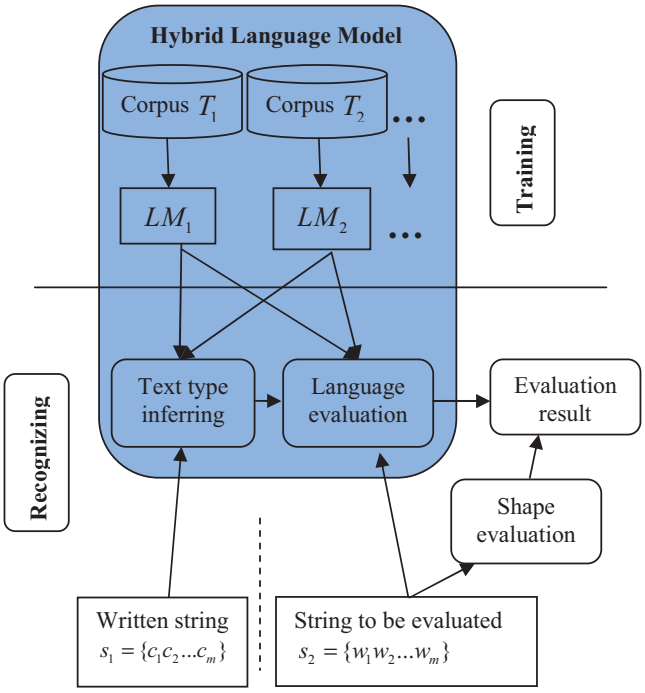

Figure 2. Prototype of the Hybrid Language Model

2.2.1. Model construction

Formally, we collect K types (denoted by $T_1, T_2, ..., T_K$) of corpus (for example, T_1 for news, T_2 for literature, T_3 for academic paper, etc.), and build K training sets (with the same size N) respectively. Then for each training set, we build a language model. So totally we get K *bi*-gram language models, denoted by $LM_1, LM_2, ..., LM_K$. Obviously, LM_i models the language behavior for the type T_i.

And also we make a new training set by selecting the same amount of corpus from the K training sets respectively and combining them together to represent a general type of corpus (denoted by T_{K+1}). Then we build a new language model LM_{K+1}. This general model, LM_{K+1}, represents the language behavior with no type bias.

2.2.2. Text type inferring

We can infer which type of text the user is writing based on the string which he has already written.

Suppose the user has written a string $s_1 = \{c_1 c_2 ... c_m\}$. We have the following equation to represent the probability of s_1 belonging to T_i:

$$P(s_1 \in T_i) = \frac{C_{T_i}(s_1)}{\sum_{i=1}^{K+1} C_{T_i}(s_1)} \quad (3)$$

where $C_{T_i}(s_1)$ denotes the count of string s_1 in the training set for T_i.

In (3), if the numerator and denominator are divided by N simultaneously, we can get:

$$P(s_1 \in T_i) = \frac{C_{T_i}(s_1)/N}{\sum_{i=1}^{K+1} C_{T_i}(s_1)/N} = \frac{p_{LM_i}(s_1)}{\sum_{i=1}^{K+1} p_{LM_i}(s_1)} \quad (4)$$

where $p_{LM_i}(s_1)$ is the probability of s_1 given by LM_i and N is the size of each training set.

Typically for *bi*-gram language models, (4) can be rewritten as

$$P(s_1 \in T_i) \approx \frac{\prod_{j=1}^{m} p_{LM_i}(c_j \mid c_{j-1})}{\sum_{i=1}^{K+1} \prod_{j=1}^{m} p_{LM_i}(c_j \mid c_{j-1})} \quad (5)$$

where $p_{LM_i}(c_j \mid c_{j-1})$ represents the conditional probability of $(c_{j-1}c_j)$ given by LM_i.

2.2.3. Language score calculation

Now we can calculate the language score based on the K+1 language models and our inferring for the text type.

Suppose a string $s_1 = \{c_1 c_2 ... c_m\}$ has already been written and recognized, and a new sentence is being recognized. And $s_2 = \{w_1 w_2 ... w_m\}$ is one candidate path of the lattice built for the new sentence. The language score of s_2 can be calculated as the weighted sum of probabilities given by different language models.

$$P(w_1 w_2 ... w_k) = \sum_{i=1}^{K+1} P(s_1 \in T_i) \cdot P_{LM_i}(w_1 w_2 ... w_k) \quad (6)$$

In the case of *bi*-gram language models, (6) can be re-written as equation (7).

$$P(w_1 w_2 ... w_k) = \sum_{i=1}^{K+1} P(s_1 \in T_i) \cdot \prod_{j=1}^{k} p_{LM_i}(w_j \mid w_{j-1}) \quad (7)$$

3. Parameter Selection

In this section, we introduce the motivation and the realization of our proposed candidate re-ranking process after DP recognition, as well as the automatic parameter selection based on Genetic Algorithm.

3.1. Candidate re-ranking

According to equation (2), the value of α, which is the combining weight of shape and language scores, is critical to the final results. A higher value of α indicates more emphasis on language behavior, while a lower value indicates more emphasis on shape characteristics.

Ideally we need a small value of α because shape recognition should be more important in a handwriting recognition engine. But in a pruning-based DP recognition system (designed for saving computational resources), such little emphasis on language behavior would lead to the cut-off of some linguistically correct paths before they are fully evaluated. This would sometimes lead to the absence of the correct string in the final recognition candidates (as illustrated in Figure 3-a). On the other hand, if we use consistently a high value of α, the shape characteristics would contribute too little to the recognition engine to produce a correct result (as illustrated in Figure 3-b).

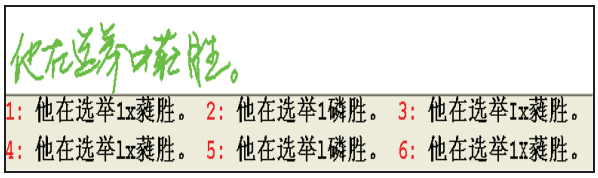

Figure 3-a. Recognition error caused by a small α

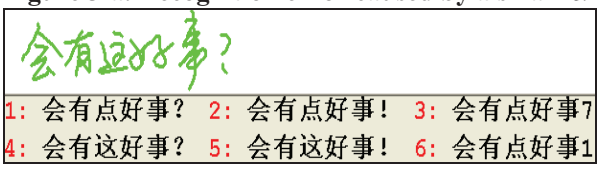

Figure 3-b. Recognition error caused by a large α

So to solve this problem, we propose to use a large value of α to maintain the linguistically correct paths during DP process, and then a small value of α to re-rank the candidates during post-operation process to make the top candidate consistent with the shape recognition. Equations (8) and (9) give the evaluations of the string $(w_1 w_2 ... w_m)$ in DP and post-operation process respectively, where the value of α_1 should be larger than that of α_2.

$$S(w_1 w_2 ... w_m) = S_{shape}(w_1 w_2 ... w_m) + \alpha_1 P(w_1 w_2 ... w_m) \quad (8)$$

$$S(w_1 w_2 ... w_m) = S_{shape}(w_1 w_2 ... w_m) + \alpha_2 P(w_1 w_2 ... w_m) \quad (9)$$

3.2. Parameter selection

We use Genetic Algorithm to select the proper parameters using the training data of handwriting sentences to maximize the recognition rate of the engine on the training data. Each parameter (i.e., α_1, α_2) is treated as a gene, an element of a chromosome. The fitness of a chromosome is given by the recognition rate on the training data using the gene of that chromosome. We initialize the chromosomes and find the best chromosome through following steps:

(1) Initialization: Generate M (population size) chromosomes with random values of each gene (from 0 to 2). Calculate the fitness of each chromosome, and the average fitness f_{ave}. Set the generation count G=0.

(2) Crossover: Select two chromosomes at a probability P_c and cross the genes at a random position to produce two new chromosomes. So we obtain W_1 new chromosomes.

(3) Mutation: Change each gene of each chromosome at a probability P_m with a random value from 0 to 2. So we obtain W_2 new chromosomes.

(4) Selection: Evaluate the fitness of the newly obtained $W_1 + W_2$ chromosomes. Select N out of $N + W_1 + W_2$ chromosomes at a probability to each chromosome which is proportional to its fitness. Calculate the average fitness of the new population f_{ave_new}. Set $G = G + 1$.

(5) Iteration: Set $f_{ave} = f_{ave\_new}$, and go to (2) for iteration except one of the following occurs:

a) $G > G_{\max}$.

b) $abs(f_{ave} - f_{ave\_new}) < threshold$ occurs n_{stop} times.

(6) End: When iteration ends, return the chromosome with the highest fitness.

It is interesting to note that, as will be demonstrated in the later experiment section, although we don't setup any constrains to guarantee $\alpha_1 > \alpha_2$ during the whole parameter selection process, it turns out that this is true ($\alpha_1 > \alpha_2$) for the best chromosome.

4. Experiments and Results

To demonstrate and validate our proposed method, in this section we present our experiments and results. We conduct 3 experiments (the details of which will be introduced following): parameter selection by GA, performance comparison between our proposed hybrid language model and a standard language model, and performance comparison between the engines with and without a re-ranking post-operation process.

4.1. Experiment setup and performance evaluation

To train the hybrid language model, we collected two types of corpus (news and literature) from the Internet. So totally we build 3 language models in our experiments (one for the news, one for the literature, and one for the general text).

To evaluate the performance, we collect 30 handwritten short articles (15 from newspapers and 15 from short-story-collection) as the benchmark. Each of the articles contains about 30 sentences, and 150 ~ 300 characters.

We recognize each article sentence by sentence. The recognized sentences are used to infer the document type and contribute to recognition the rest sentences.

The recognizing accuracy of the engine in each testing set is evaluated in character level, given by:

$$Sr = \frac{C_{correct\_charater}}{C_{total\_character}} \times 100\% \qquad (10)$$

where $C_{correct\_charater}$ is the number of characters that are correctly recognized in the top candidate, and $C_{total\_character}$ is the character count in the testing set.

4.2. Parameter selection

As described in section 3.2, we use Genetic Algorithm to choose the parameters α_1, α_2 automatically. We set population size $N = 50$, the probability of mutation $P_m = 0.15$, the probability of crossover $P_c = 0.8$, maximum generation $G_{\max} = 1000$, and the stop criteria $n_{stop} = 20$. The optimized parameters are shown below in Table 1, where DPS stands for Dynamic Programming Stage, and POS stands for Post-Operation Stage.

This result coincides with and confirms our idea that the linguistic knowledge contributes differently to the recognition engine at different stages to get better recognition results. And this provides the evidence that it is necessary to re-rank the candidates after DP recognition process with a smaller language score weight.

Table 1. The Optimized Parameters Given by GA

Parameter:	α_1	α_2
Description:	The weight of LM in DPS	The weight of LM in POS
Value:	2.34	1.56

4.3. Performance of the hybrid language model

To compare the performance between our proposed hybrid language model with the standard language model, we build the following testing sets:

T_N : The 15 handwritten articles from the newspapers.

T_L : The 15 handwritten articles from the short-story-collection.

T_{Mix} : All the 30 handwritten articles.

We build 3 recognition engines, one with the language model trained from the news corpus (denoted by LM_N), one with the language model trained from the literature corpus (denoted by LM_L) and one with the hybrid language model. And we use these three engines to recognize the above 3 testing sets respectively, and record the recognizing accuracy as Table 2. We can see from the table that, as expected, LM_N and LM_L are only effective with T_N and T_L respectively, while the hybrid language model consistently achieves good recognizing performance. More significantly, the overall performance of the hybrid model is much better than that of both LM_N and LM_L.

Table 2. Comparison of Recognizing Accuracy

	LM_N	LM_L	Hybrid Model
T_N	0.88	0.83	0.85
T_L	0.81	0.87	0.86
T_{Mix}	0.82	0.83	0.85

4.4. Performance of the post-operation process

We also evaluated the performance of the post-operation process. We build two recognition engines both with the hybrid language model, one with a re-ranking process after DP recognition (denoted by $S_{re-rank}$) and one without (denoted by S). We use these two engines to recognize T_{Mix}, and record the accuracy as Table 3. We can see clearly in the table that, the re-ranking of the candidates by a lower language score is a necessary process and improves the recognition accuracy.

Table 3. Comparison of Recognizing Accuracy between $S_{re-rank}$ and S

	S	$S_{re-rank}$
T_{Mix}	0.85	0.87

5. Conclusion

In this paper, we have proposed a hybrid language model for handwritten Chinese sentence recognition. The hybrid model integrates together and takes advantages of several language models for different corpus types. It can contribute more precisely to the recognition engine based on the prior inferring of the text type. Our experiments have shown that, the recognition engine with this hybrid model performs consistently well in different types of testing set, and gets a much better overall performance than the engine with a single standard language model.

We also proposed a candidate re-ranking process after the DP recognition. The reason is that it is necessary to re-rank the candidates by a smaller language score. By doing so, according to the experiment results, we can reach 2% performance increasing in terms of recognition accuracy.

6. References

[1] N.Furukawa, J. Tokuno, H. Ikeda, "Online character segmentation method for unconstrained handwriting strings using off-stroke features," Proc. 10th *International Workshop on Frontiers in Handwriting Recognition*, 2006

[2] T. Fukushima, M. Nakagawa, "On-line writing-box-free recognition of handwritten Japanese text condidering character size variations," Proc. 15th *International Conference on Pattern Recognition* (ICPR 00), vol.2, 2000, pp.359-363

[3] L.Y. Tseng, R.C. Chen, "Segmenting handwritten Chinese characters based on heuristic merging of stroke bounding boxes and dynamic programming," *Pattern Recognition Letter*, vol.19(10), 1998, pp. 963-973.

[4] S. Zhao, Z. Chi, P. Shi, H. Yan, " Two-stage segmentation of unconstrained handwritten Chinese characters," *Pattern Recognition*, vol.36(1) ,2003, pp:145-156

[5] S. Jaeger, C.L. Liu and M. Nakagawa, "The state of the art in Japanese online handwriting recognition compared to techniques in western handwriting recognition", *International Jounral on Document Analyasis and Recognition*, vol.6(2), 2003,pp.75-88

[6] X.D. Zhou, J.L. Yu, C.L. Liu, T. Nagasaki, K. Marukawa, " Online handwritten Japanese character string recognition incorporating geometric context," Proc. 9th *International Conference on Document Analysis and Recognition* (ICDAR07), 2007, pp.48-52

[7] B. Zhu, X.D. Zhou, C.L. Liu, M. Nakagawa, "," *International Journal on Document Analysis and Recognition*, vol.13(2), 2010, pp.121-131

[8] 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(1), 2008, pp. 167-182

[9] M. Cheriet, N. Kharma, C.L. Liu, C.Y. Suen, "*Character Recognition Systems: A Guide for Students and Practitioners*," Wiley, New York, 2007

[10] H. Murase, "Online recognition of free-format Japanese handwrittings, " Proc. 9th *International Conference on Pattern Recognition*, vol.2, 1988, pp.1143-1147

[11] P.K. Wong, C. Chan, "Postprocessing statistical language models for handwritten Chinese character recognizer," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 1999, vol.29(2), pp.286-91

[12] C.H. Chang, "Word class discovery for postprocessing Chinese handwriting recognition," Proc. *15th conference on Computational linguistics*, vol.2, 1994, pp.1221-1225