# PREFIX TREE BASED AUTO-COMPLETION FOR CONVENIENT BI-MODAL CHINESE CHARACTER INPUT

*Peng Liu, Lei Ma, and Frank K. Soong*

Microsoft Research Asia, Beijing, P. R. China, 100080
{pengliu, lema, frankkps}@microsoft.com

## ABSTRACT

We address the problem of character auto-completion (CAC) for predicting Chinese characters with partial, cursive handwriting input. A prefix tree decoder based CAC algorithm is proposed. The approach is based upon HMM pen trajectory modeling and radical structure of Chinese characters. Without finishing the strokes, high quality character candidates can be efficiently predicted. As a result, significant improvement of the recognition throughput can be obtained. We combine further the handwriting CAC with speech recognition candidates in the posterior sense, and come up with a flexible, rapid bi-modal Chinese character input system. The system was tested on a large Chinese corpus and shown that: more than 90% of input attempts can be correctly finished with only 50% of the whole character trajectory written.

***Index Terms—*** User interface, speech recognition, handwriting recognition, multi-modal, prefix tree

## 1. INTRODUCTION

In recent years, natural user interface such as speech and handwriting becomes more popular, with the advancement of pattern recognition technologies. Generally speaking, speech input is efficient but not robust against noise or other adverse environment in recognition. On the other hand, handwriting input is slower but more reliable. Based upon the pros and cons of the two modalities, attempts have been made to combine the two modes together for a convenient user interface [1, 2, 3, 4].

When inputting east Asian language character in isolation, which is the dominant and the most convenient input mode on hand-held devices, the advantages of such a bi-modal system becomes even more obvious. That is because many Chinese characters are too complicated to be written rapidly, and there are too many characters with similar pronunciations to be recognized reliably. Therefore, a bi-modal input system for entering east Asian languages is a good choice.

In our previous work, we show a conceptual bi-modal system for isolated Chinese characters [1] input, which adopts handwriting as a main input mode and use speech to automatically complete a partial-written character to improve throughput. However, a more primitive handwriting recognizer, which only accepts characters written on a very regulated stroke by stroke input is used.

To generalize the idea to "real" handwriting input, the major challenge is to predict meaningful character candidates based on partial online handwriting input. For example, when the user inputs a radical '氵', the system should be able to list out those characters consist of with this radical, such as '江', '河', and '湖', etc. The problem is called *character auto-completion (CAC)* and it is the main focus of this paper. It is easy to do CAC by table look up when assuming the character is written regularly in a stroke-by-stroke manner. For cursive input, we need an efficient CAC algorithm built in the online recognizer, and a decent pen trajectory statistical model to handle cursive handwriting input.

In this work, we adopt hidden Markov models (HMMs) to completely characterize cursively written characters at sub-character or radical level, which makes a prefix tree based decoding algorithm [5] a natural choice. Our CAC mechanism is built upon the evolving prefix tree in decoding and outputs CAC candidates with only marginal loss in efficiency. The algorithm provides CAC results simultaneously with the user's input. Hence, the user can select the correct candidate to accelerate the input speed. The user can say the character while writing it, and the CAC list can be fused with speech recognition result for a better accuracy.

The rest of the paper is organized as follows: In Section 2, we briefly introduce the radical hierarchy for prefix tree based decoding and HMM pen trajectory modeling; In Section 3, a CAC algorithm based on prefix tree is proposed; In Section 4, fusion strategy of the bi-modal system is described; In Section 5, the recognizers and bi-modal input system are tested on a real database; The paper is concluded in Section 6.

## 2. RADICAL BASED HMM PEN TRAJECTORY MODELING

Online modeling approach for handwritten Chinese characters, based upon the global or local feature emphasis, can be group into two categories: holistic and trajectory based approaches [6]. Obviously, the latter is more appropriate for

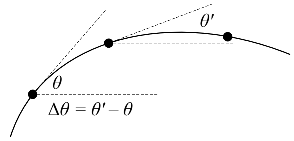


**Fig. 1**. Features for trajectory modeling

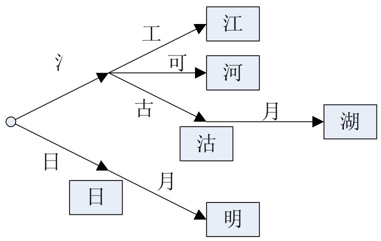


**Fig. 2**. Prefix tree for radical based of character modeling

CAC. We use HMM to model the pen trajectories with 4-dimensional features composed of slope $(\cos\theta, \sin\theta)$ and curvature $(\cos\Delta\theta, \sin\Delta\theta)$, as illustrated in Figure 1.

### 2.1. Radical based modeling and decoding

Chinese character is formed in a complex but rather hierarchical structure. Radicals or sub-character roots are commonly used as the basic semantic or phonetic construction units. This structure is similar to speech hierarchy: a word can be decomposed into one or several sub-word units, e.g. phones. With a set of radicals, we are able to represent all Chinese characters. Thus, the problem of modeling all Chinese characters becomes easier since we only need to model a relatively small number of radicals.

Once the radical set is determined, the whole radical hierarchy of all the characters can be described in a tree lexicon [5] as illustrated in Figure 2, and conventional decoding algorithms for speech recognition can be directly applied. To further improve the efficiency, beam search and histogram pruning can be used in decoding. Actually, this approach forms a natural basis for efficient CAC algorithm.

### 2.2. Multi-path HMM for stroke order modeling

A challenge problem in pen trajectory modeling is how to handle various stroke orders by different or even the same writer for the same radical. In speech recognition, Gaussian mixture models (GMMs) are widely used to automatically learn different modes in the spectrum distribution of a state. Inspired by that, we design the structure of radical HMMs in a multi-path way, as illustrated in Figure 3, and the EM training will then automatically summarize major modes of trajectory from the training data. For example, the radical '九' has two writing orders, and a two-path HMM can automatically learn these two writing sequence as shown in Figure 3.

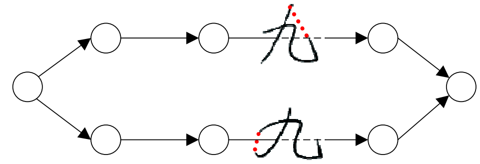

**Fig. 3**. Multi-path HMM to model different stroke orders

### 2.3. MSD HMM for imaginary stroke modeling

To facilitate HMM-based modeling, we usually connect the real strokes by *imaginary strokes* [8] to simulate strokes in pen-up phases. However, it then makes some different radicals indistinguishable, e.g., '氵 ' and '讠 '. We can treat real and imaginary strokes in different space, and the multi-space distribution (MSD) HMM [9], which has been successfully used in modeling pitch trajectory for voice/unvoice sounds, can be adopted.

In MSD modeling, the observation space $\Omega$ is considered as an union of several sub-spaces: $\Omega = \bigcup_{g=1}^{G} \Omega_g$, where $\Omega_g$ is a $n_g$-dimensional sub-space $\mathbb{R}^{n_g}$, specified by index $g$, and $G$ is the total number of sub-spaces. Each sub-space $\Omega_g$ has its probability $\omega_g$, where $\sum_{g=1}^{G} \omega_g = 1$. Correspondingly, the output probability is defined as the summation of observed probability in each sub-space: $b(\boldsymbol{o}) = \sum_{\boldsymbol{o}\in\Omega_g} \omega_g P_g(\boldsymbol{o})$, where $\boldsymbol{o} \in \Omega_g$ means that $\boldsymbol{o}$ can be observed in $\Omega_g$.

In pen trajectory modeling, the sample space $\Omega$ consists of a real sub-space $\Omega_{\mathrm{R}}$ and an imaginary sub-space $\Omega_{\mathrm{I}}$. Then the output distribution is represented as:

$$b(\boldsymbol{o}) = \begin{cases} \omega_{\mathrm{R}} P_{\mathrm{R}}(\boldsymbol{x}) & \text{if } \boldsymbol{o} \in \Omega_{\mathrm{R}} \\ \omega_{\mathrm{I}} P_{\mathrm{I}}(\boldsymbol{x}) & \text{if } \boldsymbol{o} \in \Omega_{\mathrm{I}} \end{cases} \tag{1}$$

## 3. PREFIX TREE BASED CHARACTER AUTO-COMPLETION

The problem of CAC can be described as: Given a pen trajectory, how to predict those characters that can partially best match it, which can be formulated as:

$$\hat{C} = \arg\max_{C} S(\boldsymbol{o}_1^t | \mathcal{H}_C) \tag{2}$$

where $\boldsymbol{o}_1^t$ is the observation sequence corresponding to the input trajectory piece, and $S$ represents the best partial matching score between the input observation sequence and the composite HMM $\mathcal{H}_C$ corresponding to character $C$. Usually, we want n-best instead of the top candidate as the output of CAC because of the uncertainty. Actually, this best partial matching score is defined as:

$$S(\boldsymbol{o}_1^t | \mathcal{H}) \equiv \max_{s\in\mathcal{H}} Q(\boldsymbol{o}_1^t, s) \tag{3}$$

where $Q(\boldsymbol{o}_1^t, s)$ denotes the score of best path up to time $t$ that ends in state $s$ [5], which is the variable decoding score. Obviously, directly calculating eq. (3) is exhaustive.

However, by making use of the radical hierarchy, we can significantly reduce the search cost in solving eq. (2). Note that in each radical, only the best score can survive when searching the global CAC result. We can decompose the CAC problem to radical level. First, we calculate the best partial matching score up to each radical $\mathcal{R}$ in the prefix tree:

$$S(\boldsymbol{o}_1^T|\mathcal{R}) = \max_{s\in\mathcal{R}} Q(\boldsymbol{o}_1^T, s) \tag{4}$$

Then we obtain the partial matching score for each character:

$$S(\boldsymbol{o}_1^T|\mathcal{H}) = \max_{\mathcal{R}\in\mathcal{H}} S(\boldsymbol{o}_1^T|\mathcal{R}) \tag{5}$$

By this two-step algorithm, we can efficiently calculate eq. (3): The first step of eq. (4) can be treated as a pre-processing for the whole character set, where the computational cost is in proportional to the size of the prefix tree, while in the second step of eq. (5), the computational cost is reduced to be in proportional to the average number of radicals in the character.

In practice, we only calculate the partial path scores for those paths which survive the beam search pruning. Hence, in implementing eq. (5), we visit each active radical and then propagate the score to all its descendants in the prefix tree, defined as those characters going through it. (For example, in the tree shown in Figure 2 , the descendant set of the radical '氵' is {江, 河, 沽, 湖}) It is cumbersome that size of the descendant set can be too large to be handled in a real time decoding, especially for those radicals close to the root of the tree. To alleviate this problem, we first sort all the leaf nodes, which represent output characters, by visiting the prefix tree in a depth first search (DFS). Note that in the DFS sorted character sequence, all the characters belonging to a descendant set are arranged continuously. Hence, we can represent the set by its starting and ending index $(S_{\mathcal{R}}, E_{\mathcal{R}})$.

Flowchart of the proposed CAC algorithm is listed in Table 1. It can be implemented easily in a conventional decoder with a marginal increase of its computational cost. The top candidates of the algorithm output are those predicted results assuming that character can be not finished yet. The CAC results make the combination with speech input feasible because the later is based on the speech of the whole character.

## 4. BI-MODAL FUSION WITH CAC INPUTS

Posterior based recognizer fusion has been proven successful in various applications. Given the candidate lists can corresponding scores from CAC and speech recognition, the fusion can be conducted as follows:

$$S(C) = P_{\mathrm{S}}(C|\boldsymbol{o}_{\mathrm{S}}) \cdot P_{\mathrm{H}}(C|\boldsymbol{o}_{\mathrm{H}}) \tag{6}$$

where the subscription 'S' and 'H' stands for speech and handwriting, respectively. According to the Bayesian rule, the generalized posterior term can be calculated as:

$$P(C|\boldsymbol{o}) = P^{\kappa}(\boldsymbol{o}|C) \Big/ \sum_{C'} P^{\kappa}(\boldsymbol{o}|C') \tag{7}$$

**Table 1**. Prefix tree based CAC algorithm

| |
|---|
| for $t = 1$ to $T$ |
|   **Intra-radical**: for each active radical $\mathcal{R}$ |
|     for each active state $s$ in $\mathcal{R}$ |
|       for each $s'$ succeeding $s$ |
|         $Q(\boldsymbol{o}_1^{t-1}, s) \rightarrow Q(\boldsymbol{o}_1^t, s')$<br>$S(\boldsymbol{o}_1^t|\mathcal{R}) = \max(S(\boldsymbol{o}_1^t|\mathcal{R}), Q(\boldsymbol{o}_1^t, s'))$ |
|   **Inter-radical**: for each active radical $\mathcal{R}$ (with ending state $s$) |
|     for each $\mathcal{R}'$ succeeding $\mathcal{R}$ |
|       for each entry state $s'$ in $\mathcal{R}'$ |
|         $Q(\boldsymbol{o}_1^t, s) \rightarrow Q(\boldsymbol{o}_1^t, s')$ |
|   **Predict results**: for each active radical $\mathcal{R}$ |
|     for each $C$ with index between $S_{\mathcal{R}}$ and $E_{\mathcal{R}}$ |
|       $S(\boldsymbol{o}_1^t|\mathcal{H}_C) = \max(S(\boldsymbol{o}_1^t|\mathcal{H}_C), S(\boldsymbol{o}_1^t|\mathcal{R}))$ |
|     Generate CAC N-best based on $S(\boldsymbol{o}_1^t|\mathcal{H}_C)$ |

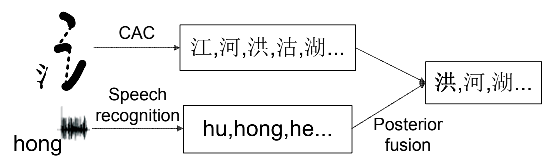


**Fig. 4**. Framework of CAC based bi-modal input system

where $\kappa$ is a scaling factor to equalized the raw candidate scores for the recognizer [7], and it is set to 0.1 in our system.

## 5. EXPERIMENTS

To evaluate our algorithm, we conducted experiments on a bi-modal, handwriting-speech database. The handwriting part is a large corpus of online Chinese handwritten characters collected on Tablet PC. The training set contains a total of 4,772,310 samples of 9,119 in printed or cursive writing styles, and the test set is composed of 355,138 samples from the same 9,119 characters. The database covers most Chinese characters currently used today and it was collected over 1,000 writers. The speech corpus is composed of 16,000 isolated syllable utterances collected from 160 speakers covers all 408 isolated syllables in Chinese. In handwriting modeling, 707 radicals are adopted and each of them is characterized by a 4-path MSD HMM. In speech modeling, we use conventional tri-phone HMMs with GMM output distributions of 6 Gaussian kernels per state.

The system setup to verify the CAC algorithm and bi-modal input is illustrated in Figure 4. We evaluate both the direct output of CAC and the bi-modal fusion results. To simulate partial handwriting input, we control the percentage of handwriting sample points sent to the decoder. In the bi-modal fusion stage, we use 500 top candidates from CAC and 20 top candidates from a speech recognizer. The recognition

**Table 2**. Character accuracies (%) of several inputs systems with partial handwriting input. (1-best/10-best)

| Percentage(%) | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100 |
|---|---|---|---|---|---|---|---|---|---|---|
| speech | 66.11/87.06 | | | | | | | | | |
| handwriting recognition | 0.02/ 0.03 | 0.02/ 0.03 | 0.02/ 0.03 | 0.03/ 0.05 | 0.06/ 0.31 | 0.28/ 2.16 | 1.90/ 8.70 | 13.26/ 44.57 | 53.26/ 79.11 | 91.84/ 97.39 |
| handwriting CAC | 0.42/ 3.80 | 1.83/ 12.12 | 5.20/ 25.98 | 12.08/ 44.81 | 24.18/ 64.40 | 39.84/ 78.67 | 56.67/ 87.76 | 71.65/ 92.80 | 82.15/ 95.15 | 86.37/ 95.91 |
| CAC + speech | 17.89/ 64.68 | 31.72/ 78.32 | 45.49/ 85.28 | 55.82/ 88.42 | 65.15/ 90.89 | 73.22/ 92.94 | 79.03/ 94.41 | 83.43/ 95.58 | 86.25/ 96.32 | 87.41/ 96.85 |

accuracies of several systems are listed in Table 2.

As expected, speech recognition accuracy of 408 isolated syllables is not so good, but the 10-best accuracy is still decent for fusion. On the other hand, even facing 9119 classes in cursive writing style, online handwriting recognition achieves decent performance with our modeling approach.

By comparing CAC and handwriting recognition results, we observe that the baseline handwriting recognition system can hardly predict the input character before it is completely finished. However, the CAC algorithm can dramatically improve hypotheses quality with partial handwriting input. Even without speech input, the throughput of handwriting input can be significantly accelerated because the user can stop input once the correct candidate appears in the 10-best list. With 50% of the whole character trajectory written, we can finish 64.40% of input attempts.

Comparing CAC and bi-modal results, we find speech input is extremely helpful to further improve the input speed. Actually, more than 90% of character input attempts can be correctly finished with only 50% of the whole character trajectory written.

It is notable that when a character is completely written , the result from handwriting recognition is still the best. Hence, we can list it with CAC results to obtain both better throughput and the best accuracy.

## 6. CONCLUSIONS AND DISCUSSIONS

We propose a prefix-based CAC algorithm to effectively predict character candidates based upon partial handwriting input. The algorithm has been implemented in a online decoder. The CAC candidate list can not only accelerate online handwriting input, but also fuse with speech recognition results. In the bi-modal system, more than 90% of character input attempts can be correctly finished with only 50%the character trajectory written.

The effectiveness of bi-modal input is based upon the radical structure of Chinese characters. The prefix tree based CAC algorithm can also be extended to western language handwriting input, where the letter decomposition of a word is treated similar to the radical radical decomposition of Chinese characters.

## 7. REFERENCES


[1] X. Zhou, Y. Tian, J. Zhou, F. K. Soong, and B. Dai, “Improved Chinese character input by merging speech and handwriting recognition hypotheses,” *Proc. ICASSP 2006*, Toulouse, France, vol. 1, pp. 609-612, 2006.

[2] P. Y. Hui and H. M. Meng, “Joint Interpretation of Input Speech and Pen Gestures for Multimodal Human-Computer Interaction,” *Proc. ICSLP 2006*, Pittsburgh, Pennsylvania, pp. 1197-1200, 2006.

[3] P. Liu, and F. K. Soong, “Word graph based speech recognition error correction by handwriting input,” *Proc. ICMI 2006*, Banff, Canada, pp. 339-346, 2006.

[4] Y. Watanabe, K. Iwata, R. Nakagawa, K. Shinoda, and S. Furui, “Semi-synchronous speech and pen input,” *Proc. ICASSP 2007*, Honolulu, Hawaii, vol. IV, pp. 409-412, 2007.

[5] H. Ney, and S. Ortmanns, “Dynamic programming search for continuous speech recognition,” *IEEE Signal Processing Magazine*, vol. 16(5), pp. 64-83, 1999.

[6] C.-L. Liu, S. Jaeger, and X.-Q. Ding, “Online recognition of Chinese characters: the state-of-the-art,” *IEEE Trans. PAMI*, vol. 26(2), pp. 198-213, 2004.

[7] W. K. Lo, F. K. SOONG, and S. Nakamura, “Generalized posterior probability for minimizing verification errors at subword, word and sentence levels”, *Proc. ISCSLP 2004*, pp.13-16, 2004.

[8] M. Okamoto, A. Nakamura, and K. Yamamoto, “Direction-change features of imaginary strokes for online handwriting character recognition,” *Pattern Recognition*, pp. 1747-1751 vol.2, 16-20 Aug. 1998.

[9] K. Tokuda, T. Mausko, N. Miyazaki, T. Kobayashi, “Multi-space probability distribution HMM,” *IEICE Trans. Inf. & Syst.*, vol.E85-D, no.3, pp.455-464, March 2002.