2009 10th International Conference on Document Analysis and Recognition

Online Handwritten Japanese Character String Recognition Using Conditional Random Fields

Xiang-Dong Zhou [1], Cheng-Lin Liu [1], Masaki Nakagawa [2]
[1] *National Laboratory of Pattern Recognition (NLPR),*
Institute of Automation of Chinese Academy of Sciences, Beijing 100190, P.R. China
[2] *Tokyo University of Agriculture and Technology, Tokyo 184-8588, Japan*
{xdzhou, liucl}@nlpr.ia.ac.cn, nakagawa@cc.tuat.ac.jp

Abstract

This paper describes an online handwritten Japanese character string recognition system based on conditional random fields, which integrates the information of character recognition, linguistic context and geometric context in a principled framework, and can effectively overcome the variable length of candidate segmentation. For geometric context, we employ both unary and binary feature functions, as well as the ones relevant and irrelevant to character classes. Experimental results show that the CRF based method outperforms the method with normalized path evaluation criterion, and the geometric context benefits the performance significantly.

1. Introduction

Character string recognition is generally accomplished by an integrated segmentation and recognition approach due to the difficulty of character segmentation [1]. Based on candidate segmentation, the information of character recognition, linguistic context and geometric context should be integrated for path evaluation. Although various criteria have been proposed [2], there still lacks a principled framework.

The path evaluation criteria can be roughly divided into two types: summation [3-5] and normalization [6,7]. The summation criterion approximates the joint probability of the candidate segmentation and its string class [4]. Since the joint probability is biased to short strings, it tends to yield over-merging errors. In [5], to overcome the effect of variable path length, each term of the probability function is weighted by the segment number of the candidate character. The normalized criteria (summations divided by the path length) are insensitive to path length, but with these criteria, the dynamic programming (DP) search cannot guarantee finding the optimal path. In [8], a conditional random field (CRF) based method is proposed for word recognition, in which the parameters are estimated by maximizing the pseudo-likelihood (product of likelihoods of cliques).

In this paper, we propose a principled maximum a posteriori (MAP) framework based on CRF [9] for fusing the information of character recognition, linguistic context and geometric context. CRFs are undirected graphical models, which can effectively capture the dependencies between characters, and provide principled tools for parameter learning and contextual recognition. Our experiments on the Kondate database [5] demonstrate the superiority of CRF based method over the normalized path criterion based method.

2. CRFs for Character String Recognition

With the maximum a posteriori (MAP) criterion, given a string pattern X, the string recognition problem is to find the optimal segmentation S^* and the associated optimal label sequence Y^* to maximize the a posteriori probability $P(S,Y|X)$:

$$(S^*, Y^*) = \arg\max_{(S,Y)} P(S,Y \mid X), \tag{1}$$

With the integrated segmentation and recognition approach, the input string X is over-segmented into a sequence of primitive segments, and between each pair of consecutive segments is a candidate segmentation point. The start and end of string pattern are viewed as two additional segmentation points. The segmentation candidate lattice describes the related set of candidate character patterns composed of consecutive segments. The path from the start to the end in the lattice represents a segmentation S of X, and Y is the label sequence of the candidate characters on S [7].

2.1. CRFs on Segmentation Candidate Lattice

Let (s,t) denote a candidate character composed of s segments between separation points t-s and t in the lattice, and (s_1,s,t) denote a pair of neighboring candidate characters $(s_1,t\text{-}s)$ and (s,t). In our CRF based method, (s,t) and (s_1,s,t) correspond to single (one-site) and pairwise (pair-site) cliques, respectively. Compared to the neighborhood graph of traditional CRF [9], in which variables are represented by nodes, the edges in segmentation candidate lattice denote candidate characters. Fig.1 shows a string pattern, its segmentation candidate lattice and corresponding neighborhood graph with nodes represent candidate characters.

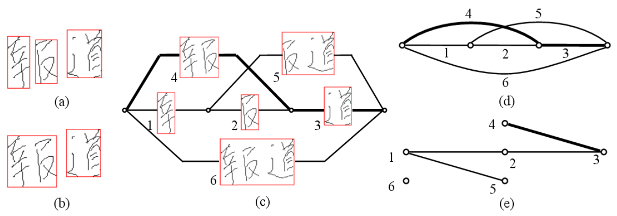

Fig. 1. (a) Primitive segments, (b) A hypothesized segmentation, (c) Candidate characters, (d) Segmentation candidate lattice, and (e) The corresponding neighborhood graph.

For a hypothesized segmentation-recognition (S,Y) of X, from the definition of CRF, $P(S,Y|X)$ can be written as the normalized product of potential functions:

$$P(S,Y\mid X)=\frac{1}{\tilde{Z}(X)}\prod_{(s_1,s,t)}\tilde{\Psi}_{(s_1,s,t)}(y_1,y) \tag{2}$$
$$=\frac{1}{\tilde{Z}(X)}\prod_{(s_1,s,t)\in S}\tilde{\Psi}_{(s_1,s,t)}(y_1,y)\prod_{(s_1,s,t)\in \bar{S}}\tilde{\Psi}_{(s_1,s,t)}(y_1,y)$$

where $\tilde{\Psi}_{(s_1,s,t)}(y_1,y)$ denotes the potential function on (s_1,s,t), and $(s_1,s,t)\in\bar{S}$ means at least one candidate character of (s_1,s,t) is not on S. y_1 and y denote the labels of $(s_1,t-s)$ and (s,t), respectively. $\tilde{Z}(X)$ is the partition function defined as:

$$\tilde{Z}(X)=\sum_{(S',Y')}\prod_{(s_1,s,t)}\tilde{\Psi}_{(s_1,s,t)}(y_1',y'). \tag{3}$$

Here we consider pairwise cliques only, and regard the potential function on single clique (s,t) as contained in the one on pairwise clique (s_1,s,t).

Note that given a segmentation path S, only the cliques on S are assumed to be characters, while the others in the lattice are assumed to be non-characters. So, if $(s_1,s,t)\in\bar{S}$, at least one of y_1 and y denotes non-character class. In Fig. 1, given the segmentation of Fig. 1(b), only the candidate patterns 4 and 3 are assumed to be characters, and all the other ones are assumed to be non-characters.

Define the potential functions on S as

$$\tilde{\Psi}_{(s_1,s,t)}(y_1,y)=\exp\left\{\sum_{k=1}^{K}\lambda_k f^k_{(s_1,s,t)}(y_1,y)\right\},\ (s_1,s,t)\in S, \tag{4}$$

where $\{f^k_{(s1,s,t)}(y_1,y)\}$ are feature functions on a pairwise clique (s_1,s,t), and $\Lambda=\{\lambda_k\}$ are weighting parameters. Since non-characters have no class labels, we define the potential functions on $\bar{S}$ as

$$\tilde{\Psi}_{(s_1,s,t)}(y_1,y)=\exp\{\lambda_0\},\ (s_1,s,t)\in\bar{S}, \tag{5}$$

where λ_0 is a parameter to be learned.

Let N_X be the number of pairwise cliques, which is a constant for a given string pattern X. Denoting the character number (path length) of a segmentation path S by $L(S)$, from Eq. (2), the a posteriori probability of (S,Y) can be written as

$$P(S,Y\mid X)$$
$$=\frac{1}{\tilde{Z}(X)}\left[\prod_{(s_1,s,t)\in S}\exp\left\{\sum_{k=1}^{K}\lambda_k f^k_{(s_1,s,t)}(y_1,y)\right\}\right]\exp\{(N_X-L(S))\lambda_0\} \tag{6}$$
$$=\frac{1}{Z(X)}\prod_{(s_1,s,t)\in S}\Psi_{(s_1,s,t)}(y_1,y)$$
$$=\frac{1}{Z(X)}\exp\{-E(\Lambda,S,Y,X)\}$$

where

$$\Psi_{(s_1,s,t)}(y_1,y)=\exp\left\{\sum_{k=0}^{K}\lambda_k f^k_{(s_1,s,t)}(y_1,y)\right\} \tag{7}$$

and $f^0_{(s1,s,t)}(y_1,y)=-1$. $\Psi_{(s1,s,t)}(y_1,y)$ is defined on $(s_1,s,t)\in S$, which means two candidate characters (s_1,s,t) are on the segmentation path and labeled as (y_1,y), and all the pairwise cliques $\{(s'_1,s',t')|t-s\le t'-s'<t\}$ belong to $\bar{S}$. $Z(X)$ is the partition function, summated over all possible combinations of segmentation and recognition:

$$Z(X)=\sum_{(S',Y')}\prod_{(s_1,s,t)\in S'}\Psi_{(s_1,s,t)}(y_1',y'). \tag{8}$$

$E(\Lambda,S,Y,X)$ is the energy function:

$$E(\Lambda,S,Y,X)=-\sum_{(s_1,s,t)\in S}\sum_{k=0}^{K}\lambda_k f^k_{(s_1,s,t)}(y_1,y) \tag{9}$$
$$=-\sum_{(s_1,s,t)\in S}\sum_{k=1}^{K}\lambda_k f^k_{(s_1,s,t)}(y_1,y)+\lambda_0 L(S)$$

which is used to measure the plausibility of (S,Y), and the smaller is $E(\Lambda,S,Y,X)$, the larger is $P(S,Y|X)$. The effect of $\lambda_0 L(S)$ is similar to [3], where the path length term is used to avoid over-merging errors in string recognition. If $\lambda_0=0$, Eq. (6) is formally equivalent to the semi-Markov conditional random field [10].

2.2. Parameter Learning

Denoting a training string sample by (S^i,Y^i,X^i) (segmentation points and character classes labeled), from the theory of energy-based model [11], training is to find the optimal parameters by minimizing a loss function, wherein the parameters are updated by stochastic gradient decent [12]:

$$\Lambda(t+1)=\Lambda(t)-\varepsilon(t)\nabla_{\Lambda}L(S^i,Y^i,X^i)|_{\Lambda=\Lambda(t)}, \tag{10}$$

where $L(S^i,Y^i,X^i)$ denotes the per-sample loss, and $\varepsilon(t)$ is the learning step. The loss function is often the negative log-likelihood (NLL) loss [11], which is a convex function and guarantees global optimization by gradient descent.

From Eq. (6), given a training sample (S^i,Y^i,X^i), to maximize $P(S^i,Y^i|X^i)$ is equivalent to minimizing $-\log P(S^i,Y^i|X^i)$. The NLL loss is thus defined as

$$\begin{aligned} L_{NLL}(S^i,Y^i,X^i) &= -\log P(S^i,Y^i \mid X^i) \\ &= E(\Lambda,S^i,Y^i,X^i) + \log Z(X^i) \end{aligned} \tag{11}$$

whose partial derivatives with respect to the weighting parameters are computed by

$$\begin{aligned} \frac{\partial L_{NLL}(S^i,Y^i,X^i)}{\partial \lambda_k} &= -\sum_{(s_1,s,t)\in S^i} f^{\,k}_{(s_1,s,t)}(y^i_1,y^i) \\ &+\sum_{(S,Y)}\sum_{(s_1,s,t)\in S} f^{\,k}_{(s_1,s,t)}(y_1,y)P(S,Y\mid X^i) \end{aligned} \tag{12}$$

Note that in the second term, the summation is over the segmentation-recognition paths and the pairwise cliques on each path. This is equivalent to the summation over all the pairwise cliques in the segmentation candidate lattice and the segmentation-recognition paths traversing each pairwise clique:

$$\begin{aligned} &\sum_{(S,Y)}\sum_{(s_1,s,t)\in S} f^{\,k}_{(s_1,s,t)}(y_1,y)P(S,Y\mid X^i) \\ &= \sum_{(s_1,s,t)}\sum_{(y_1,y)} f^{\,k}_{(s_1,s,t)}(y_1,y) \sum_{\substack{(S,Y)\in \\ \{(S,Y)\mid((s_1,s,t),(y_1,y))\in(S,Y)\}}} P(S,Y\mid X^i) \\ &= \sum_{(s_1,s,t)}\sum_{(y_1,y)} f^{\,k}_{(s_1,s,t)}(y_1,y)P((s_1,s,t),(y_1,y)) \end{aligned} \tag{13}$$

where $P((s_1,s,t),(y_1,y))$ denotes the marginal probability that (s_1,s,t) is on the segmentation path and labeled as (y_1,y). For each clique (s,t) at the start of segmentation path ($t-s=0$), $P((s_1,s,t),(y_1,y))$ is replaced by $P((s,t),y)$, which is the marginal probability that (s,t) is on the segmentation path and labeled as y.

In [9], to calculate the partial derivatives with respect to the weighting parameters, the feature functions on single and pairwise cliques are associated with unary and binary marginal probabilities, respectively. For convenience, we consider just binary probabilities (except for $t-s=0$) by regarding the feature functions on single cliques as special ones on pairwise cliques.

Similar to the linear-chain CRF [9], $P((s_1,s,t),(y_1,y))$ and $P((s,t),y)$ can be calculated from the forward variables $\alpha_{(s,t)}(y)$ and the backward variables $\beta_{(s1,t1)}(y_1)$. $\alpha_{(s,t)}(y)$ can be deduced recursively by

$$\alpha_{(s,t)}(y) = \sum_{s_1,y_1} \Psi_{(s_1,s,t)}(y_1,y)\alpha_{(s_1,t-s)}(y_1),\ (s,t)\in S \tag{14}$$

where $\alpha_{(s,s)}(y)=\Psi_{(s,s)}(y_1,y)$ and the binary feature functions on (s,s) are set as zero. $\beta_{(s1,t1)}(y_1)$ can be deduced backward:

$$\beta_{(s_1,t_1)}(y_1) = \sum_{s,y} \Psi_{(s_1,s,t_1+s)}(y_1,y)\beta_{(s,t_1+s)}(y),\ (s_1,t_1)\in S \tag{15}$$

where $\beta_{(s1,T)}(y_1)=1$ (T is the index of the last separation point). With the forward and backward variables, the partition function and the marginal probabilities are calculated by

$$Z(X) = \sum_{s,y}\alpha_{(s,T)}(y) = \sum_{s,y}\beta_{(s,s)}(y), \tag{16}$$

$$P((s,t),y) = \frac{1}{Z(X)}\alpha_{(s,t)}(y)\beta_{(s,t)}(y), \tag{17}$$

$$\begin{aligned} &P((s_1,s,t),(y_1,y)) \\ &= \frac{1}{Z(X)}\Psi_{(s_1,s,t)}(y_1,y)\alpha_{(s_1,t-s)}(y_1)\beta_{(s,t)}(y) \end{aligned}. \tag{18}$$

With $P((s,t),y)$ and $P((s_1,s,t),(y_1,y))$, the partial derivatives of the NLL loss can be calculated and used for gradient learning.

2.3. Feature Functions

In CRF, the feature functions are used to capture the node attributes and local dependencies between nodes. For character string recognition, the feature functions evaluate the scores of isolated character recognition and the compatibilities of linguistic context and geometric context. The feature functions are unary if defined on single cliques, and binary on pairwise cliques. On the other hand, the feature functions are class-relevant if dependent on character class (or class pair), otherwise are class-irrelevant. So, there are totally four types of feature functions: unary class-relevant, unary class-irrelevant, binary class-relevant and binary class-irrelevant.

The output score of character recognizer on candidate character patterns is a unary class-relevant feature function. It is given by an MQDF classifier on direction histogram features [7].

The linguistic context is measured using the logarithm of bi-gram probabilities [7], which is a binary class-relevant feature function.

On extracting unary geometric features from character patterns, a QDF classifier is trained for giving a unary class-relevant feature function. On extracting binary geometric features from pairs of character patterns, another QDF classifier (defined on super character class pairs) is trained for giving a binary class-relevant feature function. The unary geometric features are identical to those in [7]. The binary geometric features include the distance between the upper bounds, lower bounds, upper-lower bounds, lower-upper bounds, and horizontal center lines of two neighboring candidate character patterns.

To derive a unary class-irrelevant geometric feature function, the candidate character patterns are divided into two classes depending on whether a segmentation point is included (negative class) or not (positive class). An SVM classifier trained to separate two classes has its output as a unary class-irrelevant geometric feature function. The geometric features used are identical to those for unary class-relevant geometric feature functions.

To derive a binary class-irrelevant geometric feature function, geometric features are extracted from pairs of consecutive segments, which are divided into two classes depending on whether a segmentation point is included (positive class) or not (negative class). An SVM classifier trained to separate two classes has its output as a binary class-irrelevant geometric feature function. The SVM classifier inputs eight binary geometric features extracted from pairs of segments. The first five features are identical to those for binary class-relevant geometric feature functions except that pairs of character patterns are replaced by pairs of segments. The remaining three are the distance between the left bounds, right-left bounds, and vertical center lines between two consecutive segments.

The feature functions are summarized in Table 1.

Table 1. Summary of feature functions.

	Type	Features	Function
f_0	Non-character	None	-1
f_1	Unary class-relevant	Character shape	MQDF
f_2	Binary class-relevant	Bi-gram	Logarithm
f_3	Unary class-relevant	Unary geometric	QDF
f_4	Binary class-relevant	Binary geometric	QDF
f_5	Unary class-irrelevant	Unary geometric	SVM
f_6	Binary class-irrelevant	Binary geometric	SVM

3. Experimental Results

We evaluated the performance of our method in experiments on horizontal character string patterns extracted from the TUAT Kondate database, including 10,174 strings for training and 3,511 strings for test [5]. By removing the strings with pre-segmentation and candidate class selection errors (cases that correct segmentation points or character class labels are not included in the candidates), 9,053 samples were actually used for training CRF parameters.

The MQDF classifier for character recognition and the QDF classifiers for class-relevant geometric feature functions were trained on the TUAT Nakayosi database [13] and Kondate database (isolated characters in the training strings).

The two SVM classifiers for class-irrelevant geometric feature functions, with third order polynomial kernel, were trained on the samples extracted from the training strings of Kondate database.

We assume that each character has at most 6 segments. The character recognizer gives 12 candidate classes of highest scores, and the accumulated accuracy of 12 candidate classes on the characters (with correct segmentations given) of test strings is 99.07%, while the accuracy of the top rank is 86.26%.

The CRF parameters were estimated with NLL on the training strings. For comparison, the normalized path evaluation criterion [7], without the summation nature, was trained by the MCE (minimum classification error) method [14]. When training with MCE, the band in beam search [7] was 5 for normalized criterion. When testing, the beam band for normalized criterion was also 5, while CRF can use DP for path search.

In training, the string samples were processed iteratively for five cycles in stochastic gradient decent. The system was implemented in MS Visual C++ 6.0 and tested on a PC with AMD Dual Core 3800+ CPU.

Table 1 shows the effect of geometric context ($f_{3\text{-}6}$) and non-characters (f_0) on string segmentation-recognition performance without and with linguistic context (f_2), fused by CRF. F1 is the segmentation measure and CRR is the character recognition rate [5,7]. We can see that the incorporation of geometric context improves the character segmentation and recognition accuracies remarkably in the case either with or without linguistic context. On the other hand, without the class-irrelevant geometric feature functions ($f_{5\text{-}6}$), the incorporation of non-character function (with the path length term used in Eq. (9)) can improve the system performance significantly. With $f_{5\text{-}6}$, the path length term has little effect. This indicates the class-irrelevant geometric feature functions $f_{5\text{-}6}$ help resist the effect of string length variability.

Table 1. Effects of geometric feature functions ($f_{3\text{-}6}$) and non-characters (f_0). Upper part: without linguistic context; lower part: with linguistic context.

Feature functions				With f_0		Without f_0	
f_1	f_2	$f_{3\text{-}4}$	$f_{5\text{-}6}$	F1	CRR	F1	CRR
○				0.8692	0.6707	0.8280	0.6138
○		○		0.9511	0.8111	0.9031	0.7387
○			○	0.9469	0.7577	0.9576	0.7894
○		○	○	0.9743	0.8415	0.9730	0.8444
○	○			0.9341	0.8151	0.8811	0.7451
○	○	○		0.9734	0.8965	0.9605	0.8754
○	○		○	0.9641	0.8588	0.9696	0.8713
○	○	○	○	0.9862	0.9191	**0.9868**	**0.9205**

Table 2. Comparing CRF and normalized path criterion.

	Test			Training		
	F1	CRR	T(s/str)	F1	CRR	T(h)
CRF	**0.9862**	**0.9191**	0.821	0.9954	0.9538	14.38
Normalized	0.9795	0.9072	1.213	0.9916	0.9429	17.29

Table 2 lists the experimental results of CRF (with all feature functions) and the normalized path criterion on test set and training set. On test set, T is the average time cost (seconds per string), while on training set, T is the training time (hours). We can see the CRF based method outperforms the normalized path criterion based method (trained by MCE) in terms of segmentation measure and character recognition rate, and costs less training and recognition time.

With the proposed method, the string recognition errors are caused by three sources: pre-segmentation, candidate class selection and path evaluation. Path evaluation is the major concern of this paper. Pre-segmentation error refers to the case that the border between two characters is not included in the candidate segmentation points. Candidate class error refers to the case that the correct class of a character is not included in the top ranks given by character recognizer. When removing the pre-segmentation and candidate class errors (guaranteeing correct segmentation points and candidate classes), we obtained the string recognition results in Table 3. The segmentation and recognition accuracies are significantly higher than those in Table 2, but still leave a big room for improvement of path evaluation fusing linguistic and geometric contexts.

Table 3. String recognition results when guaranteeing segmentation points and candidate classes.

	F1	CRR
CRF	**0.9919**	**0.9400**
Normalized	0.9842	0.9262

Our segmentation and recognition accuracies on test set listed in Table 2 are slightly lower than those reported in [5]. This is because we use different techniques of pre-segmentation, character recognition and linguistic context. Particularly, [5] uses tri-gram for linguistic context while we use bi-gram.

4. Conclusion

We proposed a CRF based online handwritten Japanese character string recognition method, which can evaluate the candidate segmentation-recognition paths by fusing character recognition, linguistic context and geometric context in a principled maximum a posteriori (MAP) framework. In experiments on character strings of TUAT Kondate database, the CRF based method can achieve higher segmentation and recognition accuracy as well as higher efficiency compared to the normalized path evaluation based method. The results also demonstrate that the geometric context can improve the system performance significantly. We will continue to improve the pre-segmentation, character class selection and context modeling strategies for further improving the string recognition performance.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (NSFC) under grants no.60775004 and no.60825301.

References

[1] H. Murase, Online recognition of free-format Japanese handwritings, *Proc. 9th ICPR*, 1988, Vol.2, pp.1143-1147.

[2] M. Cheriet, N. Kharma, C.-L. Liu, C.Y. Suen, *Character Recognition Systems: A Guide for Students and Practitioners*, John Wiley & Sons, 2007.

[3] S. Senda, K. Yamada, A maximum-likelihood approach to segmentation-based recognition of unconstrained handwriting text, *Proc. 6th ICDAR*, 2001, pp. 184-188.

[4] M. Nakagawa, B. Zhu, M. Onuma, A model of on-line handwritten Japanese text recognition free from line direction and writing format constraints, *IEICE Trans. Information and Systems*, E88-D(8) (2005) 1815-1822.

[5] B. Zhu, X.D. Zhou, C.L. Liu, M. Nakagawa, A robust model for on-line handwritten Japanese text recognition, *Document Recognition and Retrieval XVI*, 2009.

[6] S. Tulyakov, V. Govindaraju, Probabilistic model for segmentation based word recognition with lexicon, *Proc. 6th ICDAR*, 2001, pp. 164-167.

[7] X.D. Zhou, J.L. Yu, C.L. Liu, T. Nagasaki, K. Marukawa, Online handwritten Japanese character string recognition incorporating geometric context, *Proc. 9th ICDAR*, 2007, pp. 48-52.

[8] S. Shetty, H. Srinivasan, S. Srihari, Handwritten word recognition using conditional random fields, *Proc. 9th ICDAR*, 2007, pp. 1098-1102.

[9] J. Lafferty, A. McCallum, F. Pereira, Conditional random fields: probabilistic models for segmenting and labeling sequence data, *Proc 18th ICML*, 2001, pp. 282-289.

[10] S. Sarawagi, W. Cohen, Semi-Markov conditional random fields for information extraction, *Advances in Neural Information Processing Systems*, NIPS 17, 2005, pp. 1185–1192.

[11] Y. LeCun, S. Chopra, R. Hadsell, R. Marc'Aurelio, F. Huang, A tutorial on energy-based learning. In: G. Bakir, et al. (Eds.), *Predicting Structured Data*, MIT Press, 2007.

[12] H. Robbins, S. Monro, A stochastic approximation method, Ann. Math. Stat. 22 (1951) 400–407.

[13] S. Jaeger, M. Nakagawa, Two on-line Japanese character databases in UNIPEN format, *Proc. 6th ICDAR*, 2001, pp. 566-570.

[14] B.-H. Juang, W. Chou, C.-H. Lee, Minimum classification error rate methods for speech recognition, *IEEE Trans. Speech and Audio Processing*, 5(3): 257-265, 1997.