Minimum Phone Error for Improved Discriminative Training

Michael Shell, *Member, IEEE,* John Doe, *Fellow, OSA,* and Jane Doe, *Life Fellow, IEEE*

***Abstract*— Minimum Phone Error (MPE) is a new objective function for discriminative training that has proved robust and effective in LVCSR. This paper describes the MPE objective function, gives detail blah blah optimisation ... I-smoothing (prior, impt for generaliation) other implementation issues including deweighted LMs, prob scaling, lattice boundaries & lattice size.**

***Index Terms*— Discriminative Training, MMI, MPE, Weak-sense auxiliary functions**

I. Introduction

DISCRIMINATIVE training of HMMs has a long history but has only recently begun to be used for large vocabulary continuous speech recognition (LVCSR), with a number of sites acheiving varying degrees of success with MMI training (cite....). This paper describes an objective function, Minimum Phone Error (MPE), which is designed specifically for use in continuous speech recognition; and explains the techniques which are used to optimise it.

[Summarise rest of paper with sections]

II. The MPE objective function

The objective function in MPE, which is to be maximised, is:

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r=1}^{R} \sum_{s} P_{\lambda}^{\kappa}(s|\mathcal{O}_r) A(s, s_r), \tag{1}$$

where λ represents the HMM parameters; $P_{\lambda}^{\kappa}(s|\mathcal{O}_r)$ is defined as the scaled posterior sentence probability of the sentence s being the correct one (given the model)

$$P_{\lambda}^{\kappa}(s|\mathcal{O}_r) = \frac{p_{\lambda}(\mathcal{O}_r|s)^{\kappa} P(s)^{\kappa}}{\sum_u p_{\lambda}(\mathcal{O}_r|u)^{\kappa} P(u)^{\kappa}} \tag{2}$$

where κ is a scaling factor typically less than one, $\mathcal{O}_r$ is the speech data for the r'th training sentence; and $A(s, s_r)$ is the raw phone transcription accuracy of the sentence s given the reference sentence s_r, which equals the number of reference phones minus the number of errors.

It is important to emphasise that while the MPE criterion maximises a measure of phone transcription accuracy, this is done in the context of a word recognition system: likelihoods $P(s)$ are calculated using a normal word-based language model and only valid word sequences are allowed. Given a number of competing word-level transcriptions of a sentence, the MPE criterion will try to make the more accurate transcriptions more likely; and it will measure accuracy based on how many phones are correct.

Note that although $A(s, s_r)$ is a discrete-valued function, the overall objective function is differentiable because the posterior probability $P_{\lambda}^{\kappa}(s|\mathcal{O}_r)$ is differentiable.

A. MPE compared to MCE

A natural comparison for the MPE objective function is the Minimum Classification Error (MCE) criterion [**?**], [**?**]. The MCE criterion, defined for an N-class problem (e.g. digit recognition), takes a measure of mis-recognition and embeds it in a sigmoid function; it is controlled by two parameters η and γ. If the task is isolated-phone recognition and the parameters κ, η and γ are all set to the same value, there is an equivalence between the MCE and MPE criteria (ignoring a factor $1/N$ that appears in the MCE objective function [**?**], [**?**]). This is true even though the MPE criterion contains no explicit sigmoid, because the process of weighting accuracy by sentences' posterior probabilities is an implicit sigmoid function.

MCE can be implemented for LVCSR using N-best lists [?] and lattices [?] by considering all wrong sentences as being in the wrong class. The equivalence between MPE and MCE breaks down for sentences with more than one phone. The difference is that the contributions of errors in different parts of a sentence are generally considered separately by the MPE objective function, but in MCE they will interact by affecting the gradient of the sigmoid function. Thus, MPE is not affected by whether the data is segmented into small or large segments, but the MCE objective function will be affected because it is based on sentence correctness.

III. Auxiliary functions

The theory behind the optimisation of the MPE objective function is based on weak-sense auxiliary functions. These are described as follows. It is then shown how they may be applied to MMI training, and the technique is later extended to MPE.

A. Strong- and Weak-Sense Auxiliary Functions

Let $\hat{\lambda}$ be used to represent the current model parameters and λ the parameters to be estimated. The context is an optimisation task where the objective function $\mathcal{F}(\lambda)$ is to be maximised. Let us make the following definitions:

- **Strong-sense** auxiliary function: a function $\mathcal{G}(\lambda, \hat{\lambda})$ is a strong-sense auxiliary function for a function $\mathcal{F}(\lambda)$

Manuscript received January 20, 2002; revised November 18, 2002. This work was supported by the IEEE.

M. Shell is with the Georgia Institute of Technology.

around $\hat{\lambda}$, if

$$\mathcal{G}(\lambda,\hat{\lambda})-\mathcal{G}(\hat{\lambda},\hat{\lambda})\leq\mathcal{F}(\lambda)-\mathcal{F}(\hat{\lambda}), \qquad (3)$$

where $\mathcal{G}(\lambda,\hat{\lambda})$ is a smooth function of λ. This is the standard form of auxiliary function used in expectation maximisation. Maximisation of the auxiliary is guaranteed to not decrease the value of $\mathcal{F}(\lambda)$, and hence iterative use of auxiliary functions around each new parameter estimate will find a local maximum of the function.

- **Weak-sense** auxiliary function: a function $\mathcal{G}(\lambda,\lambda')$ is a weak-sense auxiliary function for a function $\mathcal{F}(\lambda)$ *around* $\hat{\lambda}$, if

$$\left.\frac{\partial}{\partial\lambda}\mathcal{G}(\lambda,\hat{\lambda})\right|_{\lambda=\hat{\lambda}}=\left.\frac{\partial}{\partial\lambda}\mathcal{F}(\lambda)\right|_{\lambda=\hat{\lambda}}. \qquad (4)$$

The condition of being a weak-sense auxiliary function can be considered a minimum condition for an auxiliary function to be useful for optimisation. If the objective function has a maximum at $\hat{\lambda}$, the weak-sense auxiliary function, if it is convex, is also bound to have its maximum at $\hat{\lambda}$. However, in contrast to the strong-sense auxiliary function increasing the value of the weak-sense auxiliary does not necessarily increase the value of the original.

Despite the limitations of weak-sense auxiliary functions compared to strong-sense functions, there are advantages to their use. The primary advantage is that a weak-sense function may be specified for many situations where strong-sense functions cannot be used. As weak-sense auxiliary functions do not guarantee an increase in the original function, they are comparable to standard gradient descent techniques. However, the advantage of using a weak-sense auxiliary function is that there is no need to determine the appropriate learning rate, or use second-order statistics. The weak-sense auxiliary function may be selected so that it has a simple closed-form for the parameter estimation. Normally the weak-sense auxiliary function will need to be smoothed in some form to try to ensure that the auxiliary function is sufficiently convex which will help ensure that the value of the original function increases. Smoothing will be acheived by adding some function with its maximum at $\hat{\lambda}$ (so this will not affect the local differential).

B. Weak-sense auxiliary functions for MMIE

This section describes how a weak-sense auxiliary function may be used to optimise the MMI criterion for training HMMs and how, given the appropriate smoothing function, it yields the standard extended Baum-Welch (EBW) update rules. Considering only a single training utterance, $\mathcal{O}=\{\mathbf{o}_1,\ldots,\mathbf{o}_T\}$ and using a fixed language model[1], the MMI criterion may be expressed as

$$\mathcal{F}(\lambda)=\log p(\mathcal{O}|\mathcal{M}^{\text{num}})-\log p(\mathcal{O}|\mathcal{M}^{\text{den}}) \qquad (5)$$

where $\mathcal{M}^{\text{num}}$ and $\mathcal{M}^{\text{den}}$ are HMMs corresponding to the correct transcription (numerator term) and all possible transcriptions (denominator term) respectively. It is not possible to define a strong-sense auxiliary function for this expression, since the second term is negative. Therefore the inequality of equation (3) will no longer hold. However, it is possible to linearly combine individual weak-sense auxiliary functions to form an overall weak-sense auxiliary function, even when there is negation.

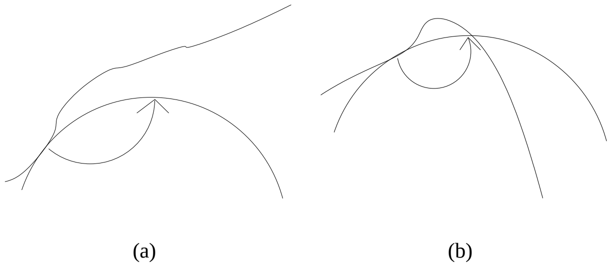
(a)
(b)

Fig. 1. Use of (a) strong-sense and (b) weak-sense auxiliary functions for function optimisation

As a strong-sense auxiliary function is by definition also a weak-sense auxiliary function, it is natural to use the standard strong-sense auxiliary function associated with ML estimation as an appropriate form for the weak-sense auxiliary function. Thus a possible weak-sense auxiliary function for the numerator term (considering a single Gaussian per state with a single dimension) is

$$\begin{aligned}\mathcal{G}^{\text{num}}(\lambda,\hat{\lambda}) &= \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_j^{\text{num}}(t)\log\left(p_\lambda(o_t|s_j)\right)\\ &= \sum_{j=1}^{J}\mathcal{Q}(\gamma_j^{\text{num}},\theta_j^{\text{num}}(\mathcal{O}),\theta_j^{\text{num}}(\mathcal{O}^2),\lambda_j) \qquad (6)\end{aligned}$$

where $\lambda_j=\{\mu_j,\sigma_j^2\}$,

$$\begin{aligned}&\mathcal{Q}(\gamma_j,\theta_j(\mathcal{O}),\theta_j(\mathcal{O}^2),\lambda_j)=\\ &\quad -\frac{1}{2}\left(\gamma_j\log(2\pi\sigma^2)+\frac{\theta_j(\mathcal{O}^2)-2\theta_j(\mathcal{O})\mu_j+\gamma_j\mu_j^2}{\sigma_j^2}\right) \qquad (7)\end{aligned}$$

s_j indicates state j of the system, $\gamma_j(t)$ is the posterior probability of being in state s_j at time t given $\hat{\lambda}$, and the sufficient statistics to evaluate the function for the numerator are given by $\theta_j^{\text{num}}(\mathcal{O})=\sum_{t=1}^{T}\gamma_j^{\text{num}}(t)o_t$, $\theta_j^{\text{num}}(\mathcal{O}^2)=\sum_{t=1}^{T}\gamma_j^{\text{num}}(t)o_t^2$ and $\gamma_j^{\text{num}}=\sum_{t=1}^{T}\gamma_j(t)$ the occupancy of the state. Similarly the auxiliary function for the denominator term alone can be defined. These two may then be subtracted to yield a candidate weak-sense auxiliary function for the MMI criterion.

As previously mentioned, in order to improve the stability of the training process, a smoothing function, $\mathcal{G}^{\text{sm}}(\lambda,\hat{\lambda})$, can be added. This may be any function with a zero differential w.r.t. λ around the current estimate $\lambda=\hat{\lambda}$. As such combining this with any weak-sense auxiliary will still be a valid weak-sense auxiliary function. Hence, for MMIE the complete weak sense auxiliary function will have the form

$$\mathcal{G}^{\text{mmi}}(\lambda,\hat{\lambda})=\mathcal{G}^{\text{num}}(\lambda,\hat{\lambda})-\mathcal{G}^{\text{den}}(\lambda,\hat{\lambda})+\mathcal{G}^{\text{sm}}(\lambda,\hat{\lambda}). \qquad (8)$$

[1] This is sometimes known as conditional maximum likelihood training.

One possible form for $\mathcal{G}^{\mathrm{sm}}(\lambda,\hat{\lambda})$ is to use D_j "effective" observations which yield the current state parameters, $\hat{\lambda}$, as the ML estimate, thus automatically satisfying the requirements for the smoothing function. This may be written in the same form as equation (6)

$$\mathcal{G}^{\mathrm{sm}}(\lambda,\hat{\lambda}) = \sum_{j=1}^{J} \mathcal{Q}(D_j, D_j\hat{\mu}_j, D_j(\hat{\mu}_j^2+\hat{\sigma}_j^2), \lambda_j), \tag{9}$$

where D_j are positive smoothing constants for each state j. The above analysis can be simply extended for multiple Gaussian components per state.

Optimising the weak-sense auxiliary function simply requires combining the sufficient statistics for each of the individual auxiliary functions. The global maximum of $\mathcal{G}^{\mathrm{mmi}}(\lambda,\hat{\lambda})$ for the mean and variance of component m of state j are given by

$$\mu_{jm} = \frac{\left\{\theta_{jm}^{\mathrm{num}}(\mathcal{O}) - \theta_{jm}^{\mathrm{den}}(\mathcal{O})\right\} + D_{jm}\hat{\mu}_{jm}}{\left\{\gamma_{jm}^{\mathrm{num}} - \gamma_{jm}^{\mathrm{den}}\right\} + D_{jm}} \tag{10}$$

$$\sigma_{jm}^2 = \frac{\left\{\theta_{jm}^{\mathrm{num}}(\mathcal{O}^2) - \theta_{jm}^{\mathrm{den}}(\mathcal{O}^2)\right\} + D_{jm}(\hat{\sigma}_{jm}^2+\hat{\mu}_{jm}^2)}{\left\{\gamma_{jm}^{\mathrm{num}} - \gamma_{jm}^{\mathrm{den}}\right\} + D_{jm}} - \mu_{jm}^2 \tag{11}$$

where D_{jm} is set on a per-Gaussian level as described in [**?**] and determines the convergence-rate and stability of the update rule. These are the standard update rules obtained from the extended Baum-Welch (EBW) algorithm [**?**], though derived using weak-sense auxiliary functions. Similarly, update equations may also be derived for the component priors and transition probabilities.

IV. Incorporating Prior Information

In this section the incorporation of a prior into the weak-sense auxiliary function framework is discussed. The derivation of I-smoothing and discriminative MAP based on MMI (MMI-MAP) and MPE (MPE-MAP) is described.

By definition, any function is both a weak and strong-sense auxiliary function of itself around any point. Thus it is possible to add any form of log prior distribution over the model parameters to a weak-sense auxiliary function and still have a weak-sense auxiliary function for a MAP version of the original function. Adding a log-prior to the MMI criterion yields

$$\mathcal{F}(\lambda) = \log p(\mathcal{O}|\mathcal{M}^{\mathrm{num}}) - \log p(\mathcal{O}|\mathcal{M}^{\mathrm{den}}) + \log p(\lambda) \tag{12}$$

The extra term can be directly added to the associated weak-sense auxiliary function leading to

$$\mathcal{G}(\lambda,\hat{\lambda}) = \mathcal{G}^{\mathrm{mmi}}(\lambda,\hat{\lambda}) + \log p(\lambda). \tag{13}$$

The exact form of the log-prior distribution affects the nature of the MAP update. One of the major issues, and choices, in MAP estimation is how to obtain this prior distribution.

A. I-smoothing

I-smoothing for discriminative training [**?**] may be regarded as the use of a prior over the parameters of each Gaussian, with the prior being based on the ML statistics. The log prior likelihood is defined as

$$\log p(\lambda_{jm}) = \mathcal{Q}\left(\tau^I, \tau^I\frac{\theta_{jm}^{\mathrm{num}}(\mathcal{O})}{\gamma_{jm}^{\mathrm{num}}}, \tau^I\frac{\theta_{jm}^{\mathrm{num}}(\mathcal{O}^2)}{\gamma_{jm}^{\mathrm{num}}}, \lambda_{jm}\right) \tag{14}$$

This log-prior is the log-likelihood of τ^I points of data with mean and variance equal to the numerator (correct model) mean and variance. Adding this log likelihod to the objective function and solving for the maximum, the MMIE update formula for the mean is then

$$\mu_{jm} = \frac{\left\{\theta_{jm}^{\mathrm{num}}(\mathcal{O}) - \theta_{jm}^{\mathrm{den}}(\mathcal{O})\right\} + D_{jm}\hat{\mu}_{jm} + \tau^I\mu_{jm}^{\mathrm{ml}}}{\left\{\gamma_{jm}^{\mathrm{num}} - \gamma_{jm}^{\mathrm{den}}\right\} + D_{jm} + \tau^I} \tag{15}$$

and for the variance,

$$\sigma_{jm}^2 = \frac{\left\{\theta_{jm}^{\mathrm{num}}(\mathcal{O}^2) - \theta_{jm}^{\mathrm{den}}(\mathcal{O}^2)\right\} + D_{jm}(\hat{\sigma}_{jm}^2+\hat{\mu}_{jm}^2) + \tau^I({\mu_{jm}^{\mathrm{ml}}}^2 + {\sigma_{jm}^{\mathrm{ml}}}^2}{\left\{\gamma_{jm}^{\mathrm{num}} - \gamma_{jm}^{\mathrm{den}}\right\} + D_{jm} + \tau^I} \tag{16}$$

where $\mu_{jm}^{\mathrm{ml}} = \frac{\theta_{jm}^{\mathrm{num}}(\mathcal{O})}{\gamma_{jm}^{\mathrm{num}}}$ and ${\sigma_{jm}^{\mathrm{ml}}}^2 = \frac{\theta_{jm}^{\mathrm{num}}(\mathcal{O}^2)}{\gamma_{jm}^{\mathrm{num}}} - {\mu_{jm}^{\mathrm{ml}}}^2$.

I-smoothing can also be directly implemented by altering the numerator statistics [**?**]. A similar form of prior with MPE training yields I-smoothing for MPE.

V. MPE

The lattice-based implementation of MPE which will be described as follows is very similar to the implementation of MMI described in Chapter **??**. Changes to the training algorithm are required at the stage at which statistics are accumulated from the training data, but the update equations are unchanged. A similar amount of computing resources are required as for MMI[2].

This chapter is organised as follows. Section VI explains an approximate method used to optimise the MPE objective function; Section VII-F discusses a more exact implementation, and Section VIII discusses the use of I-smoothing for MPE. Experimental results for MPE are given in Chapter **??**.

[Must change this.]

In order to make the optimisation efficient with lattices and with probability scaling, a constraint is made on the objective function. The posterior probability $P_\lambda^\kappa(s|\mathcal{O}_r)$ is defined as

$$P_\lambda^\kappa(s|\mathcal{O}_r) = \frac{p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa} \tag{17}$$

This involves scaled acoustic probabilities $p_\lambda(\mathcal{O}_r|s)^\kappa$. There are two options in applying this scaling. One is to apply it to whole-sentence likelihoods

$$p_\lambda(\mathcal{O}_r|s)^\kappa = \sum_X p_\lambda(\mathcal{O}_r|s,X)^\kappa$$

[2] in both cases, about 0.5× real-time per iteration of training for a typical training setup on the Switchboard corpus when run on Pentium III processors at 850 MHz.

(which is what is given in the formulation above) The other is to apply it to individual state-sequence likelihoods:

$$p_\lambda(\mathcal{O}_r|s)^\kappa = (\sum_X p_\lambda(\mathcal{O}_r|s,X))^\kappa$$

(which is recommended by schluter...) The first approach is preferred.

A technique which is in effect very close to the first approach (named “exact-match”) gave a slightly lower WER on a Switchboard experiment and had less problems with overtraining (see .... ??). The first approach (scaling whole-sentence likelihoods) is felt to be more appropriate because it does not lead to inappropriate state alignments being chosen. Note from expts (will show) that difference in err rate is larger when full-search is done.

An approximation to the first approach is to use the phone-alignments in the lattice. This is not exact because i) The lattice may contain ¿1 phone alignments for a particular sentence, which are added unscaled.
ii) State sequences not consistent with the given phone alignment may contribute a significant probability to the given sentence likelihood.

But it’s very efficient.

VI. Optimisation of the MPE objective function

The EB update formulae were developed for the optimisation of the MMI objective function, and were originally proved for that case. The same approach is not directly applicable to MPE. In the MPE objective function, which is given as follows in an expanded form,

$$\mathcal{F}_{\rm MPE}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa}, \tag{18}$$

the scales $A(s,s_r)$ applied to each sentence likelihood in the numerator are not necessarily positive, and the individual fractions are added together rather than multiplied as in MMI. This makes it difficult to derive the EB equations in the original way, as in [**?**], [?].

The solution used here is to use an intermediate weak-sense auxiliary function, based on a sum over phone arcs. The lattice for each training file r is composed of phone arcs $q = 1 \ldots Q_r$, each with given start and end times s_q and e_q (the training-file index r is omitted for brevity).

For each phone arc q, the acoustic likelihood of the speech data from the beginning to the end of the arc can be calculated; let this be called $p(q)$.

The weak-sense auxiliary function used to make the MPE objective function more tractable is:

$$\mathcal{H}_{\rm MPE}(\lambda,\lambda') = \sum_{r=1}^{R}\sum_{q=1}^{Q_r} \left.\frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}\right|^{(\lambda=\lambda')} \log p(q) \tag{19}$$

This function $\mathcal{H}_{\rm MPE}(\lambda,\lambda')$ is a weak-sense auxiliary function for $\mathcal{F}_{\rm MPE}(\lambda)$ around $\lambda = \lambda'$, for the following reason: the only change in $\mathcal{F}_{\rm MPE}(\lambda)$ as λ is changed comes via the sentence likelihoods $p_\lambda(\mathcal{O}_r|s)$, and the only variables in these that vary with λ are the arc likelihoods $p(q)$. An approximation to $\mathcal{F}_{\rm MPE}(\lambda)$ which is linear in the values of $\log p(q)$ will have the same differential w.r.t λ where $\lambda = \lambda'$, and will therefore be a weak-sense auxiliary function for $\mathcal{F}_{\rm MPE}(\lambda)$ around $\lambda = \lambda'$.

The value $\frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}|^{(\lambda=\lambda')}$ is a scalar value calculated for each arc q, and can be either positive or negative. The two cases can be separated, making the analogy with MMI clearer:

$$\begin{aligned}\mathcal{H}_{\rm MPE}(\lambda,\lambda') = & \textstyle\sum_{r=1}^{R}\sum_{q=1}^{Q_r} \max(0, \frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}|^{(\lambda=\lambda')}) \log p(q) \\ - & \textstyle\sum_{r=1}^{R}\sum_{q=1}^{Q_r} \max(0, -\frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}|^{(\lambda=\lambda')}) \log p(q)\end{aligned} \tag{20}$$

where the first term corresponds to the numerator model in MMI and the second to the denominator. As for MMI, two sets of accumulated statistics are stored: one for the numerator, and one for the denominator. (For the Gaussian updates the two sets of statistics may be compressed by saving only their difference). For I-smoothing with MPE, a third set of statistics, the “mle” statistics, are stored. The “mle” statistics are standard statistics as used for Maximum Likelihood estimation, and are obtained from a lattice forward-backward algorithm.

The final auxiliary function (without the smoothing term added yet) is:

$$\mathcal{G}_{\rm MPE}(\lambda,\lambda') = \textstyle\sum_{r=1}^{R}\sum_{q=1}^{Q_r} \frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}|^{(\lambda=\lambda')} \mathcal{G}_{\rm MLE}(\lambda,\lambda',r,q) \tag{21}$$

if $\mathcal{G}_{\rm MLE}(\lambda,\lambda',r,q)$ is understood to be the normal ML auxiliary function for the arc likelihood $p(q)$ for arc q from lattice r, as would be used for E-M training of HMMs. This expression is a weak-sense auxiliary function for Equation 19, and is therefore a weak-sense auxiliary function for the MPE objective function.

In storing statistics for updating the MPE objective function, an important definition is:

$$\gamma_q^{\rm MPE} = \frac{1}{\kappa}\frac{\partial \mathcal{F}_{\rm MPE}}{\partial \log p(q)}, \tag{22}$$

which is the differential of the objective function w.r.t the arc log likelihood $\log p(q)$, for the phone arc q, scaled by $\frac{1}{\kappa}$ which is an arbitrary scale introduced for consistency with MMI and to simplify the calculation of $\gamma_q^{\rm MPE}$.

Once this value $\gamma_q^{\rm MPE}$ is calculated, the statistics needed to optimise the objective function can easily be calculated. In a modification of Equations (**??**) to (**??**) for MMI, the numerator and denominator statistics are accumulated according to the following equations for the numerator:

$$\gamma_{jm}^{\rm num} = \sum_{q=1}^{Q}\sum_{t=s_q}^{e_q} \gamma_{qjm}(t)\max(0,\gamma_q^{\rm MPE})) \tag{23}$$

$$\theta_{jm}^{\rm num}(\mathcal{O}) = \sum_{q=1}^{Q}\sum_{t=s_q}^{e_q} \gamma_{qjm}(t)\max(0,\gamma_q^{\rm MPE})\mathcal{O}(t) \tag{24}$$

$$\theta_{jm}^{\rm num}(\mathcal{O}^2) = \sum_{q=1}^{Q}\sum_{t=s_q}^{e_q} \gamma_{qjm}(t)\max(0,\gamma_q^{\rm MPE})\mathcal{O}(t)^2, \tag{25}$$

and as follows for the denominator:

$$\gamma^{\text{den}}_{jm} = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \max(0, -\gamma^{\text{MPE}}_q)) \qquad (26)$$

$$\theta^{\text{den}}_{jm}(\mathcal{O}) = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \max(0, -\gamma^{\text{MPE}}_q)\mathcal{O}(t) \qquad (27)$$

$$\theta^{\text{den}}_{jm}(\mathcal{O}^2) = \sum_{q=1}^{Q} \sum_{t=s_q}^{e_q} \gamma_{qjm}(t) \max(0, -\gamma^{\text{MPE}}_q)\mathcal{O}(t)^2, \qquad (28)$$

where s_q and e_q are the start and end times of arc q, and $\gamma_{qjm}(t)$ are the occupation probabilities for Gaussians conditional on the arc being q. This follows from Equation (20), in which arcs with positive γ^{MPE}_q are considered as numerator arcs, and arcs with negative γ^{MPE}_q are considered as denominator arcs. The same proof used to show that the MMI objective function can be optimised with an auxiliary function $\mathcal{G}_{\text{MPE}}(\lambda, \lambda')$ of the form given in Equation (**??**), applies to the function $\mathcal{H}_{\text{MPE}}(\lambda, \lambda')$ of Equation (20) which is has the same form as the MMI objective function. The final auxiliary function $\mathcal{F}(\lambda, \lambda')$ is the same for MPE as MMI and the EB update equations are applied in the same way. The only difference with MPE is that the statistics are accumulated according to Equations 23 to 28 rather than as for MMI, the "num" and "den" being accumulated from the numerator and denominator lattices respectively as they would be for normal (scaled) ML training.

VII. Differentiating the MPE objective function

To calculate the statistics for MPE it is necessary to calculate the scaled differential γ^{MPE}_q of the MPE criterion w.r.t. the log acoustic likelihood for each arc.

Three alternative methods have been used for this: "approximate-accuracy MPE" which uses an approximation to a phone's contribution to the sentence correctness, "approximate-error MPE," which uses an approximation to a phone's contribution to the sentence error, and "exact MPE" which is an (almost) exact calculation of the differential of the MPE objective function. The methods used in so-called "exact MPE" will not be described here in detail (see [?]).

At some point the differentiation of the MPE objective function will involve implicitly calculating the function $A(s)$ for each sentence in the lattice. Before explaining the method used to differentiate the overall objective function, the approximations used to calculate $A(s)$ will be described.

A. *Calculating $A(s, s_r)$ for approximate-accuracy MPE*

The function $A(s)$ for a sentence s ideally equals the number of correct phones minus the number of insertions, but an approximation may be used to avoid the need for a full alignment. The exact form of the function (i.e., the number of correct phones minus insertions) could equivalently be expressed as a sum of $A(q)$ over all phones q in s, where $A(q)$ is defined as follows:

$$A(q) = \left\{ \begin{array}{l} 1 \text{ if correct phone} \\ 0 \text{ if substitution} \\ -1 \text{ if insertion} \end{array} \right\}. \qquad (29)$$

This expression is valid because $\#ref = (\#corr + \#sub + \#del)$ and $\#err = (\#sub + \#ins + \#del)$ so $\#ref - \#err = \#corr - \#ins$

Since the computation of the expression in Equation 29 requires alignment of the reference an hypothesis sequences, and this is computationally expensive, an approximation is used as follows. Given a hypothesis phone q, a phone z is found in the reference transcript which overlaps in time with q; and if the proportion of the length of z which is overlapped is $e(q, z)$,

$$A(q) = \max_z \left\{ \begin{array}{l} -1 + 2e(q,z) \text{ if z and q same phone} \\ -1 + e(q,z) \text{ if different phones} \end{array} \right\}. \qquad (30)$$

This is efficient to compute because it is a purely local function of the hypothesis and reference phones. The phone z is chosen so as to make $A(q)$ as large as possible. The expressions in Equation (30) represent tradeoffs between an insertion and a correct phone or substitution respectively, and are a solution to the problem that a single reference phone might be used more than once by a hypothesis sentence. In this implementation the reference phone z is chosen from a lattice encoding alternate alignments of the correct sentence. The expression in Equation (30) can be shown never to exceed the ideal value of Equation (29) provided the reference transcript has a single time alignment, i.e ignoring the fact of there being alternate paths in the reference lattice. The reference lattice may have multiple paths due to alternate pronunciations of words.

This approximation is easy to implement in a lattice context and seems to give good results. Note that unless indicated otherwise all experiments use context-free phones for purposes of calculating phone accuracy, as opposed to matching the contexts as well, since this has been found experimentally to be the best approach (Section **??**). $A(q)$ is set to zero if q is a silence phone, which is is the logical approach if silences are deleted prior to measuring the phone error. However silences are left in the reference transcription so they can be the reference phone z in the expression for $A(q)$. This appears to work slightly better [?] (it may slightly reduce the errors due to inserted phones).

B. *Example of approximate-accuracy MPE*

Figure 2 gives an example of calculating approximated RawPhoneAccuracy for a single hypothesis and reference transcript. The calculated value (0.85) is compared to the exact value (1) and is slightly less. With a single reference transcript, the approximation will always be less than or equal to the true value.

Figure 3 shows the same process in a lattice context. The correct transcription may contain alternate paths if alternate pronunciations appear in the dictionary; the reference phone z may be chosen from any path to maximise the phone accuracy. The hypothesis/recognition lattice (middle) is shown with the function $A(q)$ indicated for each phone arc.

The bottom lattice in Figure 3 shows the differential of the MPE objective function w.r.t. the log likelihoods of the different phone arcs (assuming the three paths in the lattice

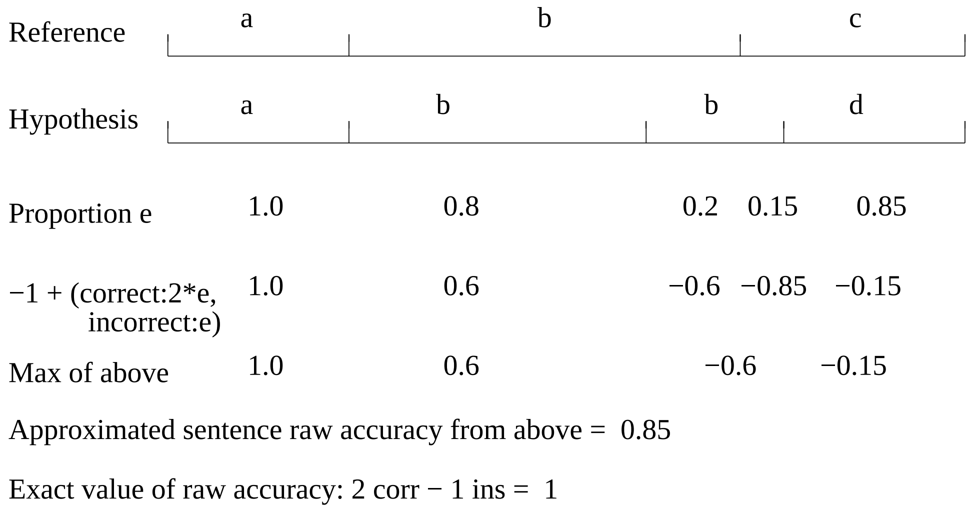
Reference
a
b
c
Hypothesis
a
b
b
d
Proportion e
1.0
0.8
0.2
0.15
0.85
−1 + (correct:2*e, incorrect:e)
1.0
0.6
−0.6
−0.85
−0.15
Max of above
1.0
0.6
−0.6
−0.15
Approximated sentence raw accuracy from above = 0.85
Exact value of raw accuracy: 2 corr − 1 ins = 1

Fig. 2. Calculating approximated RawPhoneAccuracy for approximate-accuracy MPE

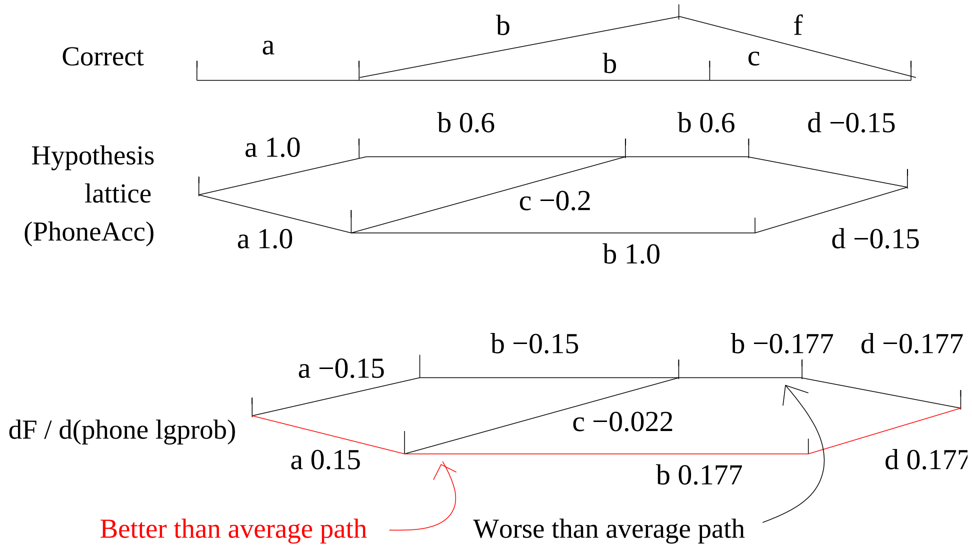

Fig. 3. Calculating approximated RawPhoneAccuracy in a lattice context

have equal likelihoods). It does not show how this differential is calculated, which is described in Section VII-E. But it makes clear a few properties of this differential, such as the fact that it is positive for paths that are more correct than average and negative for less correct paths; and that it will sum to zero for all arcs crossing a particular time instant.

To summarise, the approximated phone accuracy $A(q)$ is found for each hypothesis arc q as follows: for each arc z in the reference transcript which overlaps in time with q, let $e(q, z)$ be the number of frames q and z overlap, divided by the length in frames of z. These values $e(q, z)$ are used in the expression in Equation (30) to calculate the approximated value of $A(q)$.

1) Silences in approximate-accuracy MPE: Silence and short pause models in approximate-accuracy MPE, are handled as follows: they are ignored when they appear in the hypothesis transcript by being given a A of zero, which is consistent with an accuracy based only on the non-silence phones in the transcription. However, the silence models still appear in the reference transcript where they may be used as the reference phone z when calculating $A(q)$: hypothesis phones which align to a silence phone would be counted as substitutions since the hypothesis phone q will never be silence itself. This was found to work better in terms of test-set performance than ignoring the silences and short pauses in both reference and hypothesis lattices (see experiments in Section **??**).

C. Calculating $A(s, s_r)$ for approximate-error MPE

The formulation of MPE given in Equation 18 is a slight simplification because instead of a single transcription s_r for each file r there is a lattice containing a number of alternative transcriptions employing different pronunciations. The phone accuracy will depend on which pronunciation is taken as the reference. Let there be n_r alternative transcriptions $s_r^1 \ldots s_r^{n_r}$ of each file r. The above formulation ("approximate-accuracy MPE") is approximately equivalent to replacing

$A(s, s_r)$ with the more explicit formula $\max_{n=1}^{n_r} A(s, s_r^n)$, i.e. the best accuracy for any reference sentence. The problem with this is that it gives a higher value for longer reference pronunciations, even if the number of errors is the same. For this reason an alternative technique called approximate-error MPE is proposed, based on an approximation to the phone error rather than the phone accuracy. This solves the problem of alternative reference lengths, and also seems to be a more exact approximation to the error/accuracy when the sentence length is fixed.

The approach to alternative sentences used in approximate-error MPE is to set the sentence accuracy to:

$$\max_{n=1}^{n_r} \mathrm{Length}(s_r^n) - \min_{n=1}^{n_r} \mathrm{RawPhoneError}(s, s_r^n).$$

In algorithm which differentiates the MPE criterion, only the second term in the equation (the negated phone error) needs to be considered. The reference length is added later so the criterion can be reported in terms of phone accuracy (see Section VII-E.1).

As with approximate-accuracy MPE, the expression $-\min_{i=1}^{n_r} \mathrm{RawPhoneError}(s, s_r^n)$ is expressed in terms of a sum of a local function $-\mathrm{PhoneErr}(q)$ for each phone q in the hypothesis sentence s. This makes the algorithm possible to implement in a lattice context. $-\mathrm{PhoneErr}(q)$ is an approximation to the number of insertions, deletions and substitutions there are in the time period spanned by phone q.

The individual $\mathrm{PhoneErr}(q)$

Need approximation for each phone's contribution, it's $\#sub + \#ins + \#del$, where we use the approximation:

$$\mathrm{PhoneErr}(q) = \\ \max_{n=1}^{n_r} \left\{ \begin{array}{l} q \neq \mathrm{sil} \rightarrow \left\{ \begin{array}{l} \#sub = \max(1 - c(q,n),\ 0) \\ +\#ins = \max(t(q,n) - 1,\ 0) \\ +\#del = \max(1 - t(q,n),\ 0) \end{array} \right. \\ q = \mathrm{sil} \rightarrow \#ins = t(q,n) \end{array} \right. \tag{31}$$

where $\max_{n=1}^{n_r}$ represents taking a maximum over all the sentence pronunciations represented in the reference lattice, $t(q, n)$ is the approximated total number of non-silence phones in the reference that align with q and $c(q, n)$ is the approximated number of correct phones that align with q. The total $t(q, n) \geq 0$ is found by summing, for each reference phone z in the sentence s_r^n that overlaps with q, the proportion of each phone z that overlaps with q as a fraction of the length of z. The number of correct phones $0 \leq c(q, n) \leq 1$ is the largest amount of overlap between a phone z in s_r^n that is the same phone as q, again as a fraction of the length of z. The computation of approximated error is illustrated by an example in Section VII-D.

Approximate-error MPE, like exact MPE, seems to over-penalise insertions and leads to a reduction in the testing insertion rate relative to approximate-accuracy MPE. In order to increase the testing insertion/deletion ratio, a constant factor I is introduced into the expressions in Equation 32, as a weight on all insertion errors:

$$\mathrm{PhoneErr}(q) = \\ \max_{n=1}^{n_r} \left\{ \begin{array}{l} q \neq \mathrm{sil} \rightarrow \left\{ \begin{array}{l} \#sub = \max(1 - c(q,n),\ 0) \\ +\#ins = I\max(t(q,n) - 1,\ 0) \\ +\#del = \max(1 - t(q,n),\ 0) \end{array} \right. \\ q = \mathrm{sil} \rightarrow \#ins = I\ t(q,n) \end{array} \right. \tag{32}$$

For instance, I might be set to 0.9 or 0.85 to reduce the pressure to remove training-set insertions.

D. Example of approximate-error MPE

Figure 4 gives an example of approximate-error MPE for a single reference and hypothesis sentence. In this case, unlike approximate-accuracy MPE, the approximation gives exactly the same answer as the exact technique. This is not always the case.

Approximate vs. exact MPE: This approximate method of calculating $A(s)$ has been compared with a more exact technique, and has been found to give slightly better test-set results than the exact technique on the Switchboard corpus (although worse for Wall Street Journal). The exact technique is described in Section VII-F. Experiments comparing the two are given in Section **??**, and show no clear difference in performance.

E. Differentiating the MPE objective function for approximate MPE

The key quantity required in MPE training is the quantity:

$$\gamma_q^{\mathrm{MPE}} = \frac{1}{\kappa} \frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)}$$

for each arc q, which is the scaled differential of the MPE objective function w.r.t. each arc log likelihood. This is analogous to to an occupation probability that would arise in ML or MMI training; if positive, it is treated for purposes of accumulating statistics as a numerator occupation probability and, if negative, a denominator occupation probability.

This quantity can be found using the formula:

$$\gamma_q^{\mathrm{MPE}} = \gamma_q (c(q) - c_{\mathrm{avg}}^r), \tag{33}$$

where γ_q is the likelihood of the arc q as derived from a forward-backward likelihood computation over the arcs, $c(q)$ is the average RawPhoneAccuracy of sentences passing through the arc q, and c_{avg}^r is the average RawPhoneAccuracy of all the sentences in the recognition lattice for the r'th training file. (All these averages are weighted by the sentence likelihood).

An example giving values of $c(q)$ is shown in Figure 5. As mentioned, $c(q)$ is the average RawPhoneAccuracy of sentences passing through the arc q, weighted by probability. This example assumes that the two alternate paths are equally likely, i.e. γ_q equals 0.5 for the top and bottom paths. Arcs on the top path have a $c(q)$ of 2, which equals the number of correct phones (3) minus 1 for one insertion error. Arcs q on the bottom path have $c(q) = 1$ because there are two errors. In this case, the expression $\gamma_q^{\mathrm{MPE}} = \gamma_q(c(q) - c_{\mathrm{avg}}^r)$ equals 0.25 for the top arcs and -0.25 for the bottom arcs, and zero for

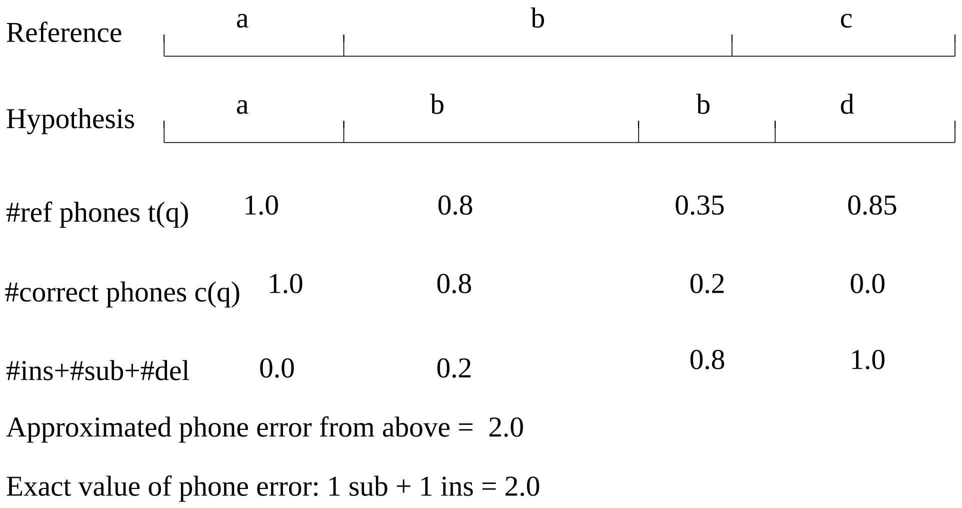

Fig. 4. Calculating approximated RawPhoneError for approximate-error MPE

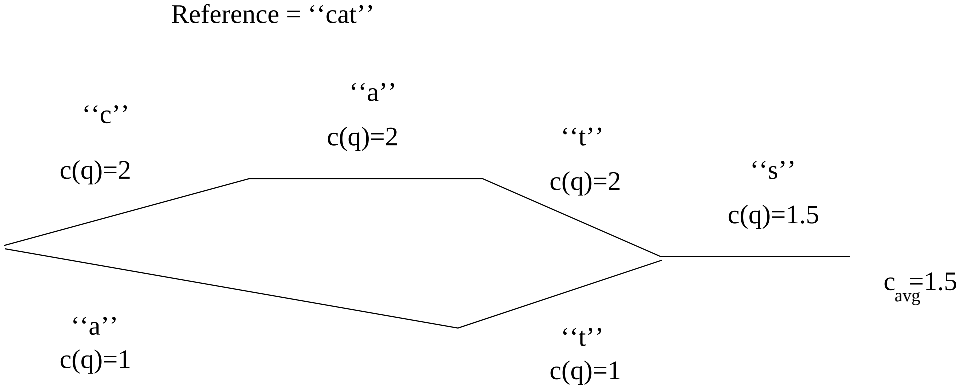

Fig. 5. Example showing values of $c(q)$ for approximate-accuracy MPE

the ending arc. These values (± 0.25) are the largest values of γ_q^{MPE} possible where the alternative sentences do not differ in correctness by more than 1. The values of γ_q^{MPE} would become smaller if the sentence were less evenly matched in likelihood.

The expression for γ_q^{MPE} given in Equation (33) can be demonstrated to be correct as follows. The MPE objective function,

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r=1}^{R} \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa},$$

can be split up into those sentences s which include a particular arc q ($q \in s$) and those which do not. Abbreviating $A(s,s_r)$ to $A(s,s_r)$,

$$\mathcal{F}_{\mathrm{MPE}}(\lambda) = \sum_{r=1}^{R} \frac{\sum_{s:q\in s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r) + \sum_{s:q\notin s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_{u:q\in u} p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa + \sum_{u:q\notin u} p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa} \quad (34)$$

Differentiating this w.r.t. the arc log likelihood $\log p(q)$ is possible by considering that for a sentence s which includes arc q ($q \in s$) the differential of its likelihood $p_\lambda(\mathcal{O}_r|s)^\kappa$ w.r.t. $\log p(q)$ equals $\kappa p_\lambda(\mathcal{O}_r|s)^\kappa$ and for other sentences ($q \notin s$) the differential of $p_\lambda(\mathcal{O}_r|s)^\kappa$ w.r.t $\log p(q)$ is zero, so by the product rule for fractions ($\frac{\partial}{\partial x}\frac{a}{b} = \frac{\partial a/\partial x}{b} - \frac{a\partial b/\partial x}{b^2}$),

$$\frac{\partial \mathcal{F}_{\mathrm{MPE}}(\lambda)}{\partial \log p(q)} = \kappa \frac{\sum_{s:q\in s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa} - \kappa \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa} \frac{\sum_{s:q\in s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa} \quad (35)$$

The expression is equal to $\gamma_q(c(q) - c^r_{\mathrm{avg}})$ of equation (33), considering that the factor κ cancels with the $\frac{1}{\kappa}$ in the definition $\gamma_q^{\mathrm{MPE}} = \frac{1}{\kappa}\frac{\partial \mathcal{F}_{\mathrm{MPE}}}{\partial \log p(q)}$, that the occupation probability γ_q equals $\frac{\sum_{s:q\in s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa}$, that the average correctness c^r_{avg} equals $\kappa \frac{\sum_s p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa A(s,s_r)}{\sum_u p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa}$ and that the average correctness of arc q equals $\frac{\sum_{s:q\in s} p_\lambda(\mathcal{O}_r|s)^\kappa P(s)^\kappa}{\sum_{u:q\in u} p_\lambda(\mathcal{O}_r|u)^\kappa P(u)^\kappa}$.

The value of $c(q)$, which is the (weighted) average value of $A(s)$ for sentences s including the phone arc q, is calculated as follows, in an algorithm similar to the forward backward algorithm.

Computation for approximate-accuracy MPE: Let the symbols α_q and β_q denote the forward and backward likelihoods used in the forward-backward algorithm to calculate occupancies $\gamma_q = \frac{\alpha_q \beta_q}{p(\mathcal{O})}$. The symbols α'_q and β'_q are used to define analogous quantities used in calculating average accuracies:

α'_q represents the average accuracy of partial phone sequences leading up to q (including q itself), and β'_q represents the average accuracy of partial phone sequences following q, so that the average accuracy $c(q)$ of phone sequences including q equals $\alpha'_q\beta'_q$. These quantities are calculated as follows:

```
for $q = 1 \ldots Q$
    if $q$ is a starting arc (no transitions to $q$)
        $\alpha_q = p(q)^\kappa$
        $\alpha'_q = A(q)$
    else
        $\alpha_q = \sum_{r \text{ preceding } q} \alpha_r t^\kappa_{rq} p(q)^\kappa$
        $\alpha'_q = \frac{\sum_{r \text{ preceding } q} \alpha'_r \alpha_r t^\kappa_{rq}}{\sum_{r \text{ preceding } q} \alpha_r t^\kappa_{rq}} + A(q)$
    end
end
for $q = Q \ldots 1$
    if $q$ is an ending arc (no transitions from $q$)
        $\beta_Q = 1$
        $\beta'_Q = 0$
    else
        $\beta_q = \sum_{r \text{ following } q} t^\kappa_{qr} p(r)^\kappa \beta_r$
        $\beta'_q = \frac{\sum_{r \text{ following } q} t^\kappa_{qr} p(r)^\kappa \beta_r (\beta'_r + A(r))}{\sum_{r \text{ following } q} t^\kappa_{qr} p(r)^\kappa \beta_r}$
    end
end

$c^r_{\text{avg}} = \frac{\sum_{\text{arcs } q \text{ at end of lattice}} \alpha'_q \alpha_q}{\sum_{\text{arcs } q \text{ at end of lattice}} \alpha_q}$

$x = \sum_{\text{arcs } q \text{ at end of lattice}} \alpha_q$
for $q = 1 \ldots Q$
    $\gamma_q = \frac{\alpha_q \beta_q}{x}$
    $c(q) = \alpha'_q + \beta'_q$
    $\gamma^{\text{MPE}}_q = \gamma_q (c(q) - c^r_{\text{avg}})$
end
```

where $A(q)$ is the (approximated) contribution of phone q to the sentence correctness, $p(q)$ is the likelihood of the data aligned to phone arc q, derived from an unscaled forward-backward probability calculation within q, t_{qr} are lattice transition probabilities derived from the language model, and the notation $\sum_{r \text{ preceding } q}$ indicates summation over phone arcs r that directly precede q in the lattice. The scaled differential w.r.t. the arc log likelihood can then be calculated according to the formula $\gamma^{\text{MPE}}_q = \gamma_q(c(q) - c^r_{\text{avg}})$. This formulation assumes that the arcs are sorted in order of time.

1) Computation for approximate-error MPE: a

For approximate-error MPE, the algorithm is as above except that $A(r)$ is replaced by $-A(r)$. The value of c^r_{avg} will now always be negative. For purposes of reporting the length-normalised MPE criterion, instead of reporting the value of $\frac{\sum_{r=1}^R c^r_{\text{avg}}}{\sum_{r=1}^R \text{Length}(s_r)}$, the value of $\frac{\sum_{r=1}^R \text{Length}(s_r) + c^r_{\text{avg}}}{\sum_{r=1}^R \text{Length}(s_r)}$ is reported, where $\text{Length}(s_r)$ is taken to be the longest of any alternative pronunciations.

F. Exact implementation of MPE

Exact MPE is covered in more detail in [?]). In exact MPE, no continuous-valued approximations to the number of phone errors in a sentence are used; an exact technique is used. There is a slight approximation involved in that a single phone arc in the lattice is not perfectly free to align simultaneously at different points in the reference transcription, as a function of the preceding and following contexts. The reference lattice, where it contains alternative pronunciations, is also approximated as a sausage. Exact MPE is significantly more complex to implement than the approximate versions described above because the calculation of phone error for sentences has to be integrated with a lattice forward-backward algorithm.

G. MPE optimisation: Summary and further details

As explained above, during the alignment (or Estimation) phase of MWE/MPE optimisation two sets of statistics are gathered: γ^{num}_{jm}, $\theta^{\text{num}}_{jm}(\mathcal{O})$ and $\theta^{\text{num}}_{jm}(\mathcal{O}^2)$ which correspond to the numerator statistics of MMI, and a similar set of statistics with the superscript "den" which correspond to the denominator statistics of MMI. These are both derived from a single lattice of recognised training data in a process which also requires the correct transcription in a phone-marked lattice form for purposes of calculating correctness of phones. The application of the technique of I-smoothing to the update process requires a third set of statistics, which are written with the superscript "mle". These are derived from the alignment of the correct utterance in the same way as for ML estimation (or for the numerator of MMI training). Transition statistics are also required to update the transition matrices; rows of the transition matrix are updated as for Gaussian weights, as described in Section **??**. The method of accumulating the transition statistics has not been described here but it is obvious by analogy with MLE training, and involves a sum over arcs very similar to the one used for Gaussian occupation counts in Equations 23 to 28.

In the algorithm described in Section VII-E, the value of $\gamma^{\text{MPE}}_q = \gamma_q(c(q) - c^r_{\text{avg}})$ is accumulated for each phone arc q in the lattice; but for reasons of efficiency, in the implementation used the sum of the value of γ^{MPE}_q over all arcs of a particular phone HMM with a particular start and end time, are added together and the arc is then treated as a single arc for purposes of accumulating data (a similar optimisation is used for calculating the within-arc likelihood $p(q)$ for such duplicated arcs). This will sometimes affect the statistics accumulated if the values of γ^{MPE}_q for that identical group of arcs differ in sign, but will not affect the fixed point of the update formula.

With MPE more iterations are generally required before the lowest WER is reached, than for MMI training. Around 8 iterations are generally required with the smoothing constant E set to the normal value of 2, as opposed to the MMI case where optimal WER may be reached after around 4 iterations. (A value of $E = 1$ or $E = 2$ is generally used for MMI).

For diagnostic purposes, the value of the MPE criterion is reported relative to the number of phones in the reference transcript. The value of c_{avg} (i.e., the MPE criterion for a given file) is summed over all files, and this value is divided by the total number of phones in all the reference transcripts (this might involve making an arbitrary decision about which reference pronunciations to use). The result will be less than 1, and is comparable to the accuracy on the training data. For exact MPE, the criterion reported is the sum of sausage lengths P plus the summed average negated errors c^r_{avg}, all divided by the sum of sausage lengths P. This will give a value between 0 and 1.

VIII. I-smoothing for MPE

I-smoothing (Section IV) is a way of obtaining smoothed estimates of discriminatively trained means and variances, using the ML statistics as the center of a prior.

The update equations with I-smoothing remain the same (Equation 15 and 16) but the Maximum Likelihood estimates μ^{ml}_{jm} and ${\sigma^{\text{ml}}_{jm}}^2$ for the mean and variance are now calculated using a separate set of statistics as for normal Maximum Likelihood training, rather than the “num” statistics which for MPE are no longer equivalent to ML statistics.

As reported in Section???, the use of I-smoothing (i.e. priors over Gaussian parameter values) is essential if MPE is to give any improvement over MMI training.

IX. Experimental results

A. Experimental setup

In order to provide results which are not too system-dependent, experiments were performed on a number of different large vocabulary corpora: the Switchboard, North American Business News (NAB, also known as Wall Street Journal), and Broadcast News (BN) corpora; and in a few cases the Resource Management corpus. Experiments on Switchboard use sets of training data of size 265 hours (h5train00), 68 hours (h5train00sub) and 18 hours (Minitrain). Broadcast News training was with a 72 hour subset of training data and NAB training used the 66 hours of channel 1 (close-talking microphone) training data. Experiments on the Resource Management corpus used the 3.8 hours of speaker-independent training data. Unless otherwise stated, experiments do not use MLLR adaptation; see Section **??** for a discussion of the interaction between MLLR and discriminative training.

Experiments use gender independent, mixture-of-Gaussian HMMs with cross-word state-clustered triphones. The input data is Mel-Frequency Perceptual Linear Prediction (MF-PLP) coefficients, which are like cepstral coefficients but derived from a process involving Mel spectrum warping and linear prediction [**?**], with delta and delta-delta coefficients, 39 dimensions in all. Resource Management experiments used standard Mel-frequency cepstral coefficients (MFCC).

There are about 6000 tree-clustered states per HMM set for the standard Switchboard, Broadcast News and NAB systems with 12 Gaussians per state for NAB, BN and the 68-hour subset of Switchboard, h5train00sub, and 16 Gaussians per state for the 265-hour h5train00 training set on Switchboard.

Testing for the main Switchboard experiments is on the 1998 DARPA Hub5 evaluation data set (eval98), about 3 hours of data; but testing for those experiments that use smaller HMM sets (identified as SW3 to SW7 in Table I) some additional tests use a subset of the 1997 evaluation test set (eval97sub) which is about 1 hour long. Error rates for NAB are given for a test consisting of both the 1994 Hub1 development and evaluation test sets. Test-set results for NAB experiments are given for the combined devalopment and evaluation from the 1994 Hub1 evaluation, 50 minutes in total. Test-set results for BN experiments are obtained using the 1996 “partitioned evaluation” development test data, which is about 2.1 hours long. Results for Switchboard and NAB are obtained with lattice rescoring of lattices obtained using ML models, for speed; results for BN are obtained using single-pass decoding.

	ID	#states	# mix	train	#frames
			/state	data (h)	/gauss
Switchboard	SW1	6165	16	265	967
Switchboard	SW2	6165	12	68	330
Switchboard	SW3	3088	6	265	5150
Switchboard	SW4	3088	6	68	1320
Switchboard	SW5	3088	6	18	350
Switchboard	SW6	3088	6	4.5	88
Switchboard	SW7	3088	6	1.125	22
WSJ/NAB	WSJ1	6399	12	66	309
WSJ/NAB	WSJ2	6399	4	66	928
WSJ/NAB	WSJ3	6399	1	66	3713
Resource Management	RM1	1582	6	3.8	144
Broadcast News	BN1	6684	12	72	323

TABLE I

Training set sizes and HMM set sizes for various setups

B. Overall comparison between MLE, MMI and MPE training

This section presents an overview of the performance gains obtainable from MPE training, as compared to ML and MMI baselines. MMI experiments are performed with and without I-smoothing; MPE experiments all use I-smoothing as it is essential in that case (see Section **??**).

The amount of performance improvement obtainable from discriminative training varies with the amount of training data and the number of Gaussians in the HMM set. Table I lists the various HMM set sizes and amounts of training data that are used in experiments reported in this section. Experiments are performed on Switchboard with varying amounts of training data and a fixed (small) HMM set size, and on the NAB database with varying numbers of Gaussians per state.

Tables II, III, and IV give the specific training setups (number of iterations, amount of I-smoothing, smoothing constant E) and improvements for the three different criteria.

Regression analysis was used to predict the relative improvement based on the amount of training data and size of the HMM set. Figure 6 shows for MPE the relative improvement predicted by the $\log_e$ (#frames of training data per Gaussian). The average squared error with this method is 9.00; adding the log (number of Gaussians) to the regression analysis as a second predictor variable only reduces this to 8.50. The #frames/Gaussian seems to predict most of the variation in relative improvement.

Figure 7 compares MPE, I-smoothed MMI and MMI for the various different corpora and HMM sets. The regression anal-

ID	E	#iters	%WER	%relative
			MLE-MMI	improvement
SW1	2	8	45.6-41.8	8.3
SW2	2	4	46.6-44.3	4.9
SW3	2	4	55.7-52.6	5.6
SW4	2	4	55.9-53.4	4.5
SW5	2	4	57.6-56.9	1.2
SW6	2	4	62.0-62.7	-1.1
SW7	2	4	77.8-80.8	-3.9
WSJ1	1	4	9.57-9.10	4.9
WSJ2	1	4	10.86-10.01	7.8
WSJ3	1	4	14.7-12.26	16.6
RM1	2	4	4.13-3.82	7.5
BN1	1	4	29.6-27.9	5.7

TABLE II
MMI TRAINING ON VARIOUS CORPORA SHOWING RELATIVE IMPROVEMENTS.

ID	E	τ^I	#iters	%WER	%relative
				MLE-MMI	improvement
SW1	1	200	8	45.6-41.4	9.2
SW2	1	200	6	46.6-43.8	6.0
SW3	2	100	4	55.7-52.8	5.2
SW4	2	100	4	55.9-54.0	3.4
SW5	2	100	4	57.6-56.1	2.6
SW6	2	100	4	62.0-61.6	0.6
SW7	2	100	4	77.8-78.2	-0.5
WSJ1	1	100	4	9.57-9.20	3.9
WSJ2	1	100	4	10.86-9.82	9.6
WSJ3	1	100	4	14.7-12.18	17.1
RM1	2	100	4	4.13-3.89	5.8
BN1	1	100	4	29.6-27.8	6.1

TABLE III
I-SMOOTHED MMI ON VARIOUS CORPORA SHOWING RELATIVE IMPROVEMENTS.

ID	E	τ^I	#iters	%WER	%relative
				MLE-MPE	improvement
SW1	2	100	8	45.6-40.8	10.5
SW2	2	50	8	46.6-43.1	7.5
SW3	1.5	50	6	55.7-50.6	9.2
SW4	1.5	50	6	55.9-52.2	6.6
SW5	1.5	50	6	57.6-54.6	5.2
SW6	1.5	50	6	62.0-60.7	2.1
SW7	1.5	50	6	77.8-79.5	-2.2
WSJ1	2	50	8	9.57-9.00	6.0
WSJ2	2	50	8	10.86-9.61	11.5
WSJ3	2	50	8	14.7-12.0	18.4
RM1	2	50	6	4.13-3.96	4.1
BN1	2	50	8	29.6-26.2	11.5

TABLE IV
MPE TRAINING ON VARIOUS CORPORA SHOWING RELATIVE IMPROVEMENTS

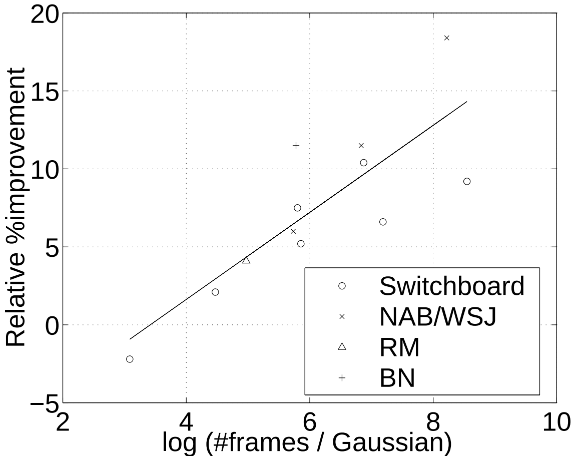

Fig. 6. MPE: relative improvements in %WER over various corpora, predicted by log(#frames/Gaussian)

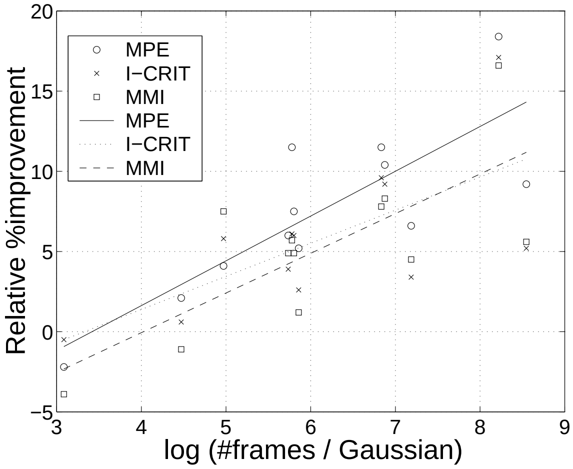

Fig. 7. MPE vs I-Crit vs MMI on various corpora

ysis shows that MPE tends to outperform MMI at all relative amounts of training data, but at very small amounts of training data I-smoothed MMI can be better than MPE. However, the point where the two meet is where the improvement is zero and there would be no point in discriminative training in any case. I-smoothed MMI is approximately the same as MMI for large amounts of training data but slightly better than MMI at small ratios.

C. *Convergence of optimisation*

Figure 8 displays the convergence of the MPE criterion (normalised by the number of words in the reference transcript), and the changes in test-set WER as training proceeds. This is for the standard HMM set sizes and training sets for the three major corpora, with the smoothing constant E set to 2 in all cases. As can be seen, there is a smooth optimisation of the criterion and a fairly smooth increase in WER. For standard MPE experiments, training is continued for 8 iterations; as can be seen from the graphs, the test-set WER is beginning to converge after 8 iterations.

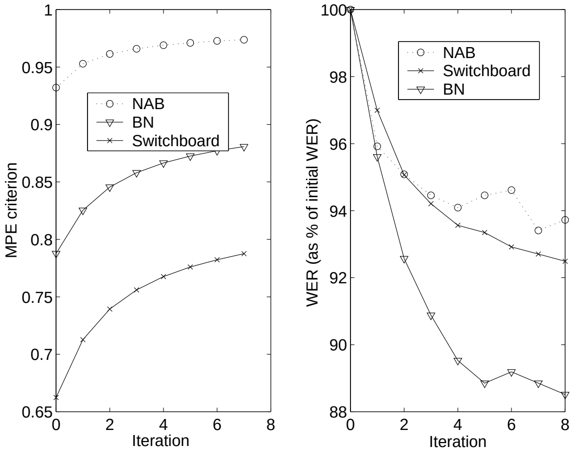

Fig. 8. Convergence of MPE criterion and WER for three corpora

D. Effect of I-smoothing constant τ^I.

τ^I	Train Subset WER		MPE Training	Test WER
	Full bg	Lat ug	Criterion	eval98
(MLE)	26.3	41.8	0.66	46.6
0	25.5	28.5	0.80	50.7
25	20.0	26.2	0.81	43.1
50	20.6	27.9	0.79	43.1
100	21.6	29.9	0.77	43.3

TABLE V

VARYING τ^I FOR I-SMOOTHING OF MPE ON SWITCHBOARD: 65H (H5TRAIN00SUB) TRAINING, E=2, 8 ITERS

Figure V (a) shows the effect of varying the I-smoothing constant τ^I for MPE training on the 68h subset of h5train00. The value of the MPE criterion (column 3) shows that higher values of τ^I (more smoothing) acts against optimisation of the MPE criterion. Training set results with the same language model used for training (unigram, column 2) show that MPE gives good training set improvements with all values of τ^I used. However, generalisation to the training set with a different language model or to the test set is very poor without any smoothing ($\tau^I = 0$), with a significant degradation in test-set performance relative to MLE. The best performance is around the range $\tau^I = 25$ to 50.

Experiments on other system setups and corpora [?] are consistent with the best value of τ^I being around 50.

E. Exact vs. ...

F. Other things...

Lattice size.. Probability scale E. bg vs ug *Exact MPE, approx ones.. mpe vs. mwe *With MLLR [?] context vs no context

mds
November 18, 2002

G. Subsection Heading Here

Subsection text here.

1) Subsubsection Heading Here: Subsubsection text here.

	MLE	Approximate accuracy MPE	Approximate error MPE			Exact MPE		
			I=0.85	I=0.9	I=1.0	I=0.85	I=0.9	I=1.0
		% WER after 8 iterations						
SW2	46.6	43.2	43.1		43.3		43.1	43.2
WSJ1	9.57	8.94		8.88	8.91		8.88	8.86
WSJ3	14.7	11.99		12.10	11.77		11.83	11.77
BN1	29.6	26.3		26.2	26.2		26.3	26.3
		ins/del ratio after 8 iterations						
SW2	0.27	0.34	0.28		0.25		0.35	0.33
WSJ1	0.95	1.07		0.95	0.92		1.10	1.06
WSJ3	0.95	0.81		0.81	0.76		0.76	0.76
BN1	0.98	1.09		0.89	0.85		1.12	1.07

TABLE VI

APPROXIMATE-ACCURACY VS. APPROXIMATE-ERROR VS. EXACT MPE ON VARIOUS CORPORA.

X. CONCLUSION

The conclusion goes here.

APPENDIX I
PROOF OF THE FIRST ZONKLAR EQUATION

Appendix one text goes here.

APPENDIX II

Appendix two text goes here.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LATEX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

PLACE
PHOTO
HERE

Michael Shell Biography text here.

John Doe Biography text here.

Jane Doe Biography text here.