PHONE DURATION MODELING FOR LVCSR

D. Povey

IBM T. J. Watson Research Renter, Yorktown Heights, N.Y. 10598
dpovey at us.ibm.com

ABSTRACT

Modeling phone durations in a word-specific fashion has previously been shown to lead to improvements in LVCSR recognition performance. We report results on the Switchboard database which confirm that at least small improvements (around 0.2-0.3% absolute) can be obtained. The duration probabilities are applied to time-marked recognition lattices. Features of the system include a novel data-driven method for smoothing discrete distributions, and a form of discrete distribution which allows phone and word lengths to be modeled simultaneously within a consistent probabilitic framework.

1. INTRODUCTION

The framework for duration modeling used here in this work is based on the approach used in [1], in which useful improvements in WER of 0.7% to 1.0% are demonstrated on the Switchboard database. That approach applied duration likelihoods to durations obtained using alignment of N-best lists; the phone lengths in a word were considered as a feature vector which was modeled using a mixture of Gaussians, and these lengths were trained separately for each word. Words were modeled separately depending on whether silence or non-silence followed, to account for the phenomenon known as pre-pausal lengthening. For unseen words the model backed off to phones and triphones.

In this work, an alternative approach based on smoothed discrete distributions is described (Section 2) and compared with the mixture-of-Gaussians approach; correlations between lengths of different phones in a word are handled by a technique described in Section 3. Prepausal lengthening is handled by a more general technique of word-context clustering (Section 4). Experiments are performed on the Switchboard database (Section 6).

2. SMOOTHED DISCRETE DISTRIBUTIONS

HMM durations in speech are difficult to model accurately with Gaussians because they have a very non-Gaussian shape, including a long tail and a sharp cutoff below the minimum duration of the HMM. This makes a discrete distribution attractive for modeling durations, i.e. having a separate $p(t)$ for each integer $0 \leq t \leq T$ for some upper limit T. The main difficulty with this approach, as with other discrete probability estimation problems (e.g. language modeling) is that of unseen symbols. An elegant solution to this problem has been devised which may also have relevance for other tasks such as language modeling.

This work was done at Cambridge University, Dept. of Engineering. Use was made of equipment donated by IBM under an SUR award.

2.1. Generalisation matrices

Suppose the task is to estimate a probability distribution over discrete symbols; without loss of generality, let these be integers $t = 1 \ldots T$ (for durations this would start at zero). Let the training data for estimating a distribution consist of a fixed number of examples $N > 0$. The naive approach would be to set $p(t) = f(t)$, where $f(t)$ is the frequency of symbol t in the training data, $f(t) = c(t)/N$, if $c(t)$ is the count of symbol t in the training sample.

A better approach can be found, if we suppose that there are in addition D example distributions $d = 1 \ldots D$, each with $N + 1$ samples drawn from the distribution. (E.g. these would tend to be distributions for a number of different classes, phones, words etc.) These distributions have counts $c_d(t)$. The idea is to use these to train a so-called "Generalisation Matrix" M of size $T \times T$ which linearly transforms the probabilities, so that an observation of symbol t gets redistributed to symbol s with weight M_{ts}. The probability $p(s)$ is now no longer $f(s)$ but $\sum_{t=1}^{T} M_{ts} f(t)$.

2.2. Estimating M

The matrix M can be estimated from the example distributions in a hold-one-out procedure where the likelihood of the held-out data is maximised. The log-likelihood of the $D(N + 1)$ example observations, based on frequencies trained on the other N examples from each distribution and transformed by M, is maximised. This objective function can be written as:

$$F(M) = \sum_{d=1}^{D} \sum_{t=1}^{T} c_d(t) \log \sum_{s=1}^{T} c'_d(s,t) M_{st}$$

where $c'_d(s,t) = c_d(s)$ if $s \neq t$ and $c_d(s) - 1$ otherwise, to take account of the held-out aspect of the training.

This expression can be maximised by E-M as follows. Starting from a flat start $M_{st}^{(0)} = 1/T$ on iteration 0, on each iteration $n > 0$ accumulate a matrix of counts

$$C_{st}^{(n)} = \sum_{d=1}^{D} c_d(t) \frac{c'_d(s,t) M_{st}^{(n)}}{\sum_{u=1}^{T} c'_d(u,t) M_{ut}^{(n)}}$$

Then the M-step of the optimisation is:

$$M_{st}^{(n+1)} = \frac{C_{st}^{(n)}}{\sum_{u=1}^{T} C_{su}^{(n)}}$$

The final value of M will tend to be closer to the identity matrix as the number of training samples N increases, because for larger N less smoothing is needed to give optimal likelihoods for held-out data.

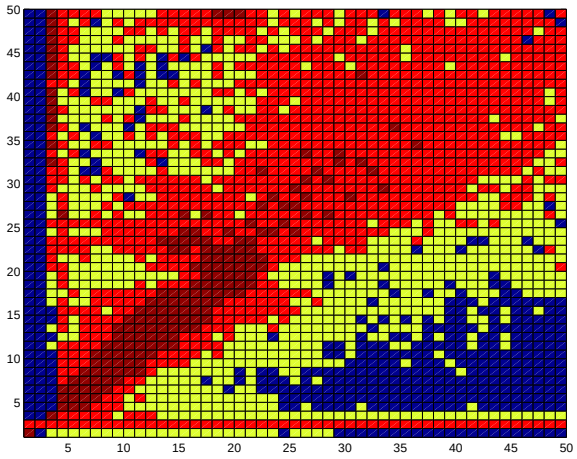

Fig. 1. Matrix M for HMM lengths of single-phone words: N=8

Note that in principle this technique requires estimating a different M for each number of training examples. In practice a different matrix M is used for each power of 2 of training examples up to some limit, and for other numbers of samples the nearest M is used from this set. In order to obtain a set of D distributions each containing $N+1$ observations, when what is available is is a different number of distributions containing a variable number of observations, a sample is taken from those of the available distributions which have at least $N+1$ observations, and exactly $N+1$ samples are randomly drawn from each of these sets of samples, possibly repeatedly.

Note that a large value of D may be needed to ensure a robust estimate of the the matrix (e.g. 100,000$/N$ is used here). The elements of the matrix with large s or t sometimes have no examples. An attempt has been made to smooth M by adding smoothed counts to the count matrix C, which leads to a more reasonable-looking M but has no appreciable effect on WER.

3. MODELING CORRELATIONS

In [1], correlations between the lengths of the phones in each word are modeled using a mixture-of-Gaussians approach. In this work the correlations are modeled as follows.

3.1. Optimising non-orthogonal distributions

Suppose the task is to estimate a probability distribution $p(\mathbf{x})$. One very general class of distributions is a product of individual functions $q_i(g)$, for $1 \leq i \leq I$:

$$p(\mathbf{x}) = q_1(g_1(\mathbf{x}))q_2(g_2(\mathbf{x})) \ldots q_I(g_I(\mathbf{x})). \quad (1)$$

where the functions $g_i(\mathbf{x})$ are scalar functions of the vector $\mathbf{x}$ which each project it onto a single dimension. The functions $q_i(g)$ are multiplied to make the final distribution but cannot be considered probabilities for non-orthogonal $g_i(\mathbf{x})$. The projections $g_i(\mathbf{x})$ are not necessarily linear functions and not necessarily orthogonal.

The optimisation of $q_i(g)$ is possible in the continuous case (assuming the projection of the current distribution can be calculated) but it is much easier to demonstrate for the discrete case. The objective is to maximise the probability of the observed data while maintaining the sum-to-one constraint of the overall distribution. Maximising the probability of the observed data in isolation would be simple once the appropriate statistics of the training data in the projection $g_i(\mathbf{x})$ are known. Maintaining the sum-to-one constraint involves knowing the projected distribution of the current distribution $p(\mathbf{x})$ with the function $g_i(\mathbf{x})$. The enforcement of the sum-to-one constraint involves more than finding a normalising factor; it also affects the shape of the distribution.

In the discrete case (i.e. where $g_i(\mathbf{x})$ is a discrete function of a discrete or continuous vector $\mathbf{x}$), the solution for $q_i(g)$ is:

$$q_i(g) = \hat{q}_i(g)\frac{f_i(g)}{p_i(g)}, \quad (2)$$

where $\hat{q}_i(g)$ is the old value of $q_i(g)$, $f_i(g)$ are the projected frequences of the training data points using $g_i(\mathbf{x})$ and, $p_i(g)$ are the projected frequencies of the current distribution $p(\mathbf{x})$ using $g_i(\mathbf{x})$.

This can be thought of as scaling a fraction $p_i(g)$ of the distribution so that it becomes the optimal fraction $f_i(g)$. Calculating and storing the data frequencies $f_i(g)$ will in general be easy. Calculating the projected distribution $p_i(g)$ of the current distribution $p(\mathbf{x})$ is more difficult, although in the special case that will be used for duration modeling it will be tractable.

3.2. Accumulated length probability model

In the accumulated length probability model (ALPM), if a word has N phones with lengths $\mathbf{l} = l_1 \ldots l_N$ there are $2N$ projections. The first N projections are the individual lengths, so $g_1(\mathbf{l}) = l_1$ etc. The last N projections are the accumulated lengths, so $g_{N+1}(\mathbf{l}) = l_1$, $g_{N+2}(\mathbf{l}) = l_1 + l_2$, etc. This allows efficient optimisation of the functions $q_i(g)$.

3.2.1. Optimisation technique

On the first iteration of optimisation, the functions $q_i(l)$ for $i = 1 \ldots N$ are set to the observed frequencies $f_i(l)$ and the other q functions are all set to 1, so $q_i(l) = 1$ for $N+1 \leq i \leq 2N$. Subsequently, each i in turn is taken and $q_i(\cdot)$ is updated, and this is repeated for a number of iterations. The functions $q_i(l)$ are updated according to Equation 2. The old functions $\hat{q}_i(l)$ are of course known; the frequencies $f_i(l)$ are accumulated from the examples of the word in the training data and stored in an array; the most complicated part is calculating $p_i(l)$. This is done as follows.

Define $\alpha_i(l)$ as the probability that the sum of phone lengths $l_1 \ldots l_i$ equals l, given only the terms $q_i(\cdot)$ and $q_{i+N}(\cdot)$ up to and including the i'th phone length (and sum of phone lengths).

For $i = 0$, $\alpha_i(l)$ is set to 1 for $l = 0$ and to 0 for $1 \leq l \leq T$. For $i > 0$ it is calculated recursively as follows:

$$\alpha_i(l) = \sum_{m=0}^{l} \alpha_{i-1}(m)q_i(l-m)q_{N+i}(l). \quad (3)$$

Note that the length l varies from 0 to T in this notation, since HMMs can have zero length in HTK.

We define $\beta_i(l)$ as the the probability that the sum of phone lengths $l_1 \ldots l_i$ equals l, but including only the terms after i. This is defined so that the α and β terms multiply to give $p_{N+i}(l)$ which is the probability of the first i phones summing to l. So the last β,

$\beta_N(l)$, equals 1, and the preceding ones are calculated as follows:

$$\beta_i(l) = \sum_{m=l}^{T} q_{i+1}(m-l) q_{N+i+1}(m) \beta_{i+1}(m). \quad (4)$$

Now, the projections of the distribution can be calculated as follows:

$$p_{N+i}(l) = \alpha_i(l)\beta_i(l) \quad (5)$$

$$p_i(l) = \sum_{m=0}^{T-l} \alpha_{i-1}(m) q_i(l) q_{i+N}(l+m) \beta_{i+1}(l+m) \quad (6)$$

where the expression for the probability $p_i(l)$ involves a sum over the length m of the lengths $l_1 \ldots l_{i-1}$. As a check, note that the equality $\sum_{l=0}^{T} p_i(l) = 1$ should hold for each $1 \leq i \leq 2N$, on each iteration.

Prior to training the ALPM, the frequencies $f_i(l)$ are smoothed as described in Section 2 When applying the update to the $q_i(\cdot)$ functions in Equation 2, a limit must be put on the factor by which the values can change, to avoid overflow and underflow. This can happen because the smoothing leads to incompatible distributions. The use of the ALPM leads to a small improvement in WER as compared with modeling the phones independently (see Section 6).

4. WORD CONTEXT CLUSTERING

In order to provide more specific modeling of words in context, each word can potentially be modeled separately for each phone context (i.e. the phones before and after the word). Since there obviously will not be enough training data for each context, the contexts are clustered. Most words will only appear in a few contexts, so it is difficult to generate a separate clustering tree for each word; hence, a global tree is used. This makes it possible to do without hand-crafted clustering questions and to use a completely data-driven approach.

A top-down clustering algorithm is performed, in which the contexts are initially all in one cluster, this cluster is split in two and then the child nodes are split recursively. For each cluster to be split, it is either split using the following phone context or the preceding phone context, whichever would give the highest increase in likelihood. The likelihood distributions are Gaussian (for speed), with a floor on the variance. When splitting a cluster according to, say, the preceding context, the preceding phones currently in the cluster are split randomly into two groups. Then for each preceding phone, the likelihood change that would result if words with that phone context were switched to the other cluster, is worked out. If positive, the phone is moved to the other cluster. Working out this likelihood change involves summing over all words, the log likelihood change that would result from modeling that word's phone lengths with differently clustered Gaussians. Note that these likelihood changes are worked out based on the assumption that the Gaussians' means and variances are recalculated when the clusters are changed, unlike conventional K-means clustering in which the objective function change is worked out based on fixed cluster centers. This difference is critical because word contexts will often be moved to a cluster that is previously empty as far as that particular word is concerned (this is possible because the tree is shared globally). This process of moving phones from cluster to cluster is repeated until there is no further change.

When using the tree to work out likelihoods for an example of a word (in a test-set word lattice), the tree is traversed from the top down to reach the node has the most specific context with more than a certain number of training examples (e.g. 200). If less than that number of examples are present at the top level for that word, the top node is used (smoothing means that even one observation will give a reasonable distribution). If there are less examples of the word than a smaller threshold (e.g. 10), the algorithm backs off to using likelihoods for each phone-in-context in the word, or, if unseen, the clustered phone-in-context as determined by the clustering used for the acoustic model.

The use of clustering gives a small improvement (about 0.2%) in WER, of which about half is obtainable with a single split in the tree. This single split roughly corresponds to prepausal lengthening– it is found that the top node splits according to the following context, with one node including silence and a few phones of the kind that occur in hesitations (um, er, etc.) and the other node including all other phones.

5. FURTHER DETAILS

Silences and short-pause models which occur at word-end in HTK, are considered as separate words for duration modeling purposes, and are modeled as separate single-phone words. Phone context is used for these words, as for normal words. As for normal phone clustering in HTK, the "short pause" model is considered invisible for purposes of calculating phone context. Experiments show that using silence contexts gives a small improvement or no difference relative to not using context. Different pronunciations of each word are considered as separate words for purposes of modeling durations.

6. EXPERIMENTS

6.1. Experimental setup

Experiments were on the Switchboard corpus, using the same Switchboard training data (360h in total) as used to train the Cambridge submission to the 2003 NIST evaluation [3]. Duration models were trained as described above on the phone-aligned transcriptions of the training data, with the maximum time T that was modeled separately being 100. Clustering of word context was performed as above, with 200 being the minimum training examples needed before a cluster-node was used in testing; below 1 example, phone-in-context probabilities were used. Testing was on the 2002 and 2003 test sets. This was done by adding duration probabilities to lattices that were generated for lattice-rescoring purposes during the 4th pass of the Cambridge system [3], and then finding the best pass through the lattices using a scale of 4 on the duration probabilities and a phone-insertion probability of 4 (to normalise the insertion/deletion ratio). The lattices used were generated with models that used MPE, HLDA, VTLN, and MLLR including lattice MLLR and full-variance transforms.

6.2. Experimental results

Table 1 shows the effect on WER of duration modeling on Switchboard. The duration modeling approach described here (marked "baseline") gives a 0.3% improvement on eval02 and 0.2% on eval03. This is disappointing when compared with the gains of up to 1.0% reported in [1] on the same corpus. Removing some of the

	%WER		log p/word	
	eval02	eval03	eval02	eval03
No durs	24.9	24.2	n/a	n/a
Durs (baseline)	24.6	24.0	-8.406	-9.21
Durs (uncorrelated)	24.8	24.1	-8.63	-9.41
Durs (phones)	25.0	24.3	-8.70	-9.48

Table 1. Duration modeling on Switchboard

	%WER		log p/word	
	eval02	eval03	eval02	eval03
No durs	24.9	24.2	n/a	n/a
Durs (baseline)	24.6	24.0	-8.41	-9.21
Durs (gauss-mix)	24.7	24.0	-9.15	-9.86

Table 2. Discrete model (ALPM) vs mixture-of-Gaussians

complexity of the system degrades likelihood and WER. Modeling the individual phones of words with indepenent discrete distributions ("uncorrelated") makes the results about 0.1-0.2% worse. The third line of the table shows the modeling where only the phone-in-context is used for duration modeling, i.e. the word identities are ignored. This slightly degrades likelihood but sharply degrades WER, which then becomes worse than doing no duration modeling. This is surprising because duration is an acoustic phenomenon, and if the normal phone clustering is sufficient for normal acoustic modeling, it ought to be sufficient for durations as well. This appears not to be the case.

Table 2 compares mixture-of-Gaussians modeling of the vectors of phone lengths in a word (as in [1]) with the discrete approach using the ALPM as described here. The number of Gaussians used was $(n/10)^{0.5}$ if there were n training examples. The Gaussian-based system used a cluster-depth of 1 (see below) in case a larger depth gave rise to generalisation problems. (Smoothed discrete modeling was used for the backoff to phones, but this probably makes little difference). Although the use of Gaussians makes the likelihoods considerably lower, the error rate is only 0.1% worse than the techniques descibed here on eval02 and identical on eval03. Since the results are so much poorer than [1], it is not clear whether the implementation of the Gaussian-based technique was very optimal.

Table 3 shows the effect of limiting the clustering depth used in clustering the contexts of words. Removing all clustering (depth=0) gives a small degradation (0.1% on both test sets) but using only one split in the clustering tree gives the same results as using the entire tree. As mentioned previously, this seems to be modeling prepausal lengthening; it splits according to the following context,

	%WER		log p/word	
	eval02	eval03	eval02	eval03
No durs	24.9	24.2	n/a	n/a
Durs (clust-depth=max)	24.6	24.0	-8.41	-9.21
Durs (clust-depth=1)	24.6	24.0	-8.39	
Durs (clust-depth=0)	24.7	24.1	-8.76	

Table 3. Effect of depth of clustering

with one set of phones being *en, em, ah* and *sil* and the other being the other 42 phones.

7. CONCLUSIONS

The experiments reported here are only moderately encouraging in terms of the effectiveness of duration modeling. The technique as implemented here will only be acceptable if an increase in system complexity is accepted for a quite small improvement in WER. However it may be useful for system combination because the transcriptions with and without duration modeling are considerably differnt (about 5%). The small gains obtained from duration modeling were only present when durations were modeled on a word rather than phone basis. It was also important to take into account word context, at least to the extent of modeling pre-pausal lengthening. There was no significant difference in WER between the techniques reported here and our implementation of the approach described in [1].

8. REFERENCES

[1] Gadde, V. R. R (2000). "Modeling Word Duration for Better Speech Recognition," Proc. Of Speech Transcription Workshop, May 16-19, 2000, Maryland.

[2] Chung G. & Seneff S (1997). "Hierarchical Duration modeling for Speech Recognition using the ANGIE framework", Speech Communication 27 (1999), 113-134.

[3] Woodland P.C. et al. (2003), "CU-HTK STT Systems for RT03", presentation to RT-03 EARS Eval Workshop, Boston, MA.