Context-Dependent Word Duration Modelling for Robust Speech Recognition

Ning Ma and Phil Green

Speech and Hearing Research Group, Department of Computer Science,
University of Sheffield, UK
{n.ma, p.green}@dcs.shef.ac.uk

Abstract

Conventional hidden Markov models (HMMs) have weak duration constraints. This may cause the decoder to produce word matches with unrealistic durations in noisy situations. This paper describes techniques for modelling context-dependent word duration cues and incorporating them directly in a multi-stack decoding algorithm. The proposed model is capable of penalising duration constraints of a word depending on its context. Experiments on connected digit recognition show that the new system can significantly improve recognition performance at different noise levels.

1. Introduction

Automatic speech recognition (ASR) based on Hidden Markov Models (HMMs) has achieved great success when applied to the problem of connected digit recognition [1], but performance often degrades significantly in the presence of noise. One reason is that conventional HMMs have unrealistic duration constraints that do not accurately reflect the true duration features encoded in speech signals [1]. When recognising speech corrupted with noise, it is easy for the decoder to produce word matches with unusual durations using models trained on clean speech. This sometimes has disastrous consequences during the matching process: word strings where the associated models have short durations tend to be favoured over competing strings with fewer words but longer durations. This effect can be observed in a connected digit recognition task with no grammar constraints, where the number of insertion errors greatly exceeds that of deletions and substitutions in noisy conditions [2].

There have been several attempts to employ explicit state duration models by adapting HMM-based systems [2, 3], but the minor improvements produced often do not justify the extra complexity introduced. Different words have different durational statistics and they are relatively insensitive to moderate noise levels, although they do obviously depend on speaking rate [4]. While the meaning of modelling state-level durations is obscure, modelling word-level duration constraints is potentially more effective for improving ASR in noise. However, modelling state duration does not necessarily produce a good model of word durations, because of the Markov state independence assumption. Our goal here is to use word duration constraints to combat the corruption of acoustic features in noisy conditions.

Within a sentence, word durations also depend on lexical stress, surrounding word and pause context. For example, long pauses affect the word durations of immediately preceding words, an effect known as 'pre-pausal lengthening' [5]. To capture these characteristics, it is necessary to model word durations depending on words context. In the next section we propose a histogram context-dependent word duration model. A multi-stack decoding algorithm is presented to directly utilise the duration model. Experiments conducted with the Aurora 2 connected digit recognition task are described in Section 4. In Section 5 results are presented and discussed.

2. Word Duration Modelling

2.1. Context-Independent Word Duration Model

In word-level HMM based systems, whole word durations are difficult to model accurately with single Gaussian because their distribution has a skewed shape. Fig. 1 shows the word duration histograms for digits 'oh' and 'six' from the Aurora database produced by forced-alignment. As word durations are themselves discrete, it makes a discrete distribution very attractive for a small vocabulary task. For a large (or even medium) vocabulary task it may become intractable to get sufficient training data for such a discrete duration model, thus a parametric model (e.g. Gaussian mixture model) may be required but can be used in a same manner. Word duration histograms were determined from an automatic Viterbi alignment for each word in the vocabulary based on a set of well-trained word-level HMMs. Word durations of about 2500 examples per digit from the training data were used to compute the histograms with a bin width of 10 ms (Fig. 1). They were then smoothed using a 5-point median filter, shown as solid lines in Fig. 2. To add a word duration model to the ASR framework, we need to estimate $P(D|w)$: the probability of word w having a duration D. To evaluate $P(D|w)$, the histograms are normalised to have area 1 so that they are equivalent to probabilities.

Because of the high dimensionality of the feature vectors typically used, we also need to introduce a scaling factor to control the duration model's impact on recognition results (like the scaling factor for a language model). This forms the word duration penalty *wdp* as:

$$wdp = P(D|w)^{\gamma} \tag{1}$$

where γ is the empirical scaling factor on word durations.

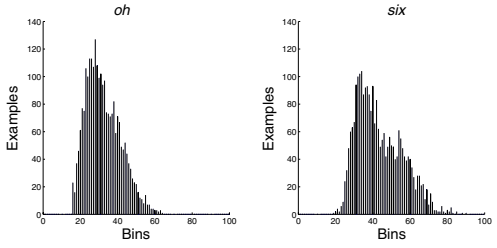

Figure 1: Raw word duration histograms of digits '*oh*' and '*six*' in the Aurora 2 training data, produced by forced-alignment.

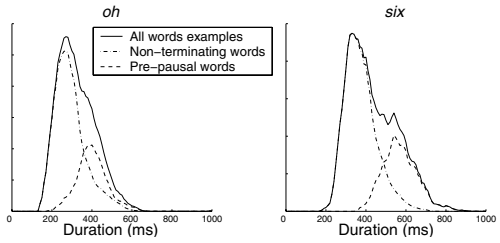

Figure 2: Word duration histograms of digits 'oh' and 'six' in the Aurora training data, produced by forced-alignment. The solid line is the smoothed word duration histogram for all word examples. The dash-dotted line represents the duration histogram of the digit when followed by a digit (non-terminating words). The dashed line is the histogram of the digit followed by a long pause (pre-pausal words).

2.2. Context-Dependent Word Duration Model

Word durations also depend on lexical stress, surrounding word or pause context. Speakers tend to lengthen a word if they want to emphasise it. Words followed by a long pause also tend to have longer durations than those followed by a word. These effects can be observed even in the connected digits domain, as shown in Fig. 2. The smoothed duration histogram (solid line) of digit 'six' has two peaks; one around 340 ms and the other around 570 ms. Furthermore, the original distribution has a very wide variance: from 160 ms to 900 ms. As in a connected digits recognition task the high-level linguistic cues are minimised, so the effect of lexical stress is not obvious. Our experiments have shown that the duration statistic of a digit spoken at the beginning or middle of an utterance is unaltered by different terminating digits. For example, in digit strings '*ONE two three*' and '*ONE three*', the two '*ONE*'s have similar duration statistics. Hence in this work we only model the 'pre-pausal lengthening' effect.

To examine the 'pre-pausal lengthening' effect, for each digit we manually divided the duration examples from the training data into two parts: examples followed by a long pause and examples followed by digits. In the Aurora 2 connected digits corpus there is long silence at the beginning and the end. We consider this silence as a long pause. Two histograms were then computed based on the two parts of duration examples. Fig. 2 shows the smoothed histograms for digits 'oh' and 'six'. The dash-dotted line represents the histogram obtained from the examples followed by a digit (non-terminating words). The dashed line is the histogram obtained from the examples followed by a long pause (pre-pausal words). We can see that the pre-pausal duration distribution has a peak at the same duration as the second peak of the original distribution. The first peak is mainly due to the non-terminating word duration distribution. Both the distributions have relatively narrower variances: 160-750 ms and 350-900 ms, respectively.

In context-dependent duration modelling we need to estimate $P(D|w_1, w_2)$: the probability of word w_1 having duration D, if it is followed by w_2. In our case w_2 can only be either pause or non-pause. For each digit the two histograms are normalised to evaluate $P(D|w_1, w_2)$. By applying the scaling factor, we can compute the context-dependent word duration penalty:

$$wdp = P(D|w_1, w_2)^{\gamma} \quad (2)$$

3. Decoding with a Word Duration Model

3.1. Multi-stack Decoding

We wish to apply word duration constraints to word sequence hypotheses as they leave word-final states but we cannot do this directly in a standard Viterbi algorithm because competing paths may have different histories, with different durations for the word now terminating. In context-dependent word duration modelling we also need to know the immediately following w_2 to compute the word duration penalty Eq. (2) for word w_1. Therefore a decoding algorithm based on multiple stacks [6, 7] is introduced. Multi-stack decoding sets up a separate stack for word sequence hypotheses that end at each time frame and processes these stacks time-synchronously from left to right. Newly created hypotheses are added to stacks and this process is continued until a complete hypothesis is determined. The items on each stack are word sequence hypotheses $H(t, W(t), P(t))$ which consist of:

1. The reference time t at which the hypothesis ends.
2. The word sequence $W(t) = w(1)w(2)\dots w(n)$ covering the time from 1 to t.
3. Its overall likelihood $P(t)$.

The decoder extracts the most likely hypothesis from the stack based on its overall likelihood at time t, computes one-word extensions, applies word duration constraints for the word, and places all the new hypotheses into corresponding stacks. When the search finishes, the most likely hypothesis path on the last stack is the optimal path.

To make the multi-stack search more efficient, some heuristic pruning can be applied to reduce the computation cost. For example, when the top hypothesis of each stack is extended for one more word w, we need only consider extensions between a minimum word duration and a maximum duration (D_{min} and D_{max}), obtained by examining word duration statistics from the training data. This word duration boundary itself seems to be able to improve the recognition performance as hypotheses with very short or very long words will be pruned out of the search. This is illustrated as a system with uniform word duration modelling in Section 4. A typical duration range for a non-terminating digit in the Aurora 2 corpus is 200-700 ms. For a pre-pausal digit a typical duration range is 300-900 ms.

3.2. Description of Algorithm

Let T denote the length of the utterance in frames and let $H(t, W^*(t), P^*(t))$ be the most likely hypothesis on the stack at time t, where $W^*(t)$ is the best word sequence finishing at time t and $P^*(t)$ is its likelihood. A Viterbi search $V(t, u, v)$ can be used to find the best-matching single words starting from a given time t and finishing at each time within a range u to v. The full algorithm uses $V(\dots)$ as follows:

1. *Initialisation*:
   Run $V(1, D_{min}, D_{max})$ to find initial one-word matches for $t = D_{min} \dots D_{max}$, and place initial hypotheses $H(t, W(t), P(t))$ in the stacks at $D_{min} \dots D_{max}$.
2. *Iteration*:
   For $t = D_{min}$ to $T - D_{min}$
   (a) Select $H(t, W^*(t), P^*(t))$ after applying word duration penalties $P(d|w^*(n), w(n+1))^{\gamma}$ from the stack at t, where d is the duration of the best-matching word $w^*(n)$ finishing at time t, and

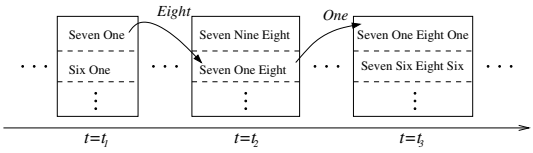

Figure 3: Illustration of the multi-stack decoding algorithm. The stack at time t_1 is being processed. 'Eight' is the best-matching single word starting at t_1 and finishing at t_2. See text for more details.

$w(n+1)$ is the next extending word. Note, with different extending words the penalty is different.

(b) Run $V(t, t+D_{min}, t+D_{max})$; form extended hypotheses and add them to each stack respectively.

3. *Termination*:
Find $H(T, W^*(T), P^*(T))$ from the stack at time T and the final result $W^*(T)$.

As we keep the best word sequence in each stack, there is no need to do backtracking to find the global optimal word sequence.

This algorithm is illustrated in Fig. 3. The most likely word sequence hypothesis ('*Seven One*') is extended by the most probable one-word extension '*Eight*' finishing at time t_2. When the decoder continues to process the stack at time t_2, a word duration penalty $P(D = t_2 - t_1 | w_1 = `eight', w_2)$ is first applied to the likelihood score of hypothesis '*Seven One Eight*', and w_2 should be decided by the next searching word. If the search goes into an HMM for silence, the penalty will be different from that if the search goes into an HMM for a digit. Since in Aurora 2 database an individual digit has a maximum duration of 900 ms (90 frames), although the search space is increased by a factor of 90, the computational load increases by a much smaller factor because most of the calculation is in the observation probability computation which does not scale up.

4. Experiments

4.1. Test Database

The experiments reported here employ the Aurora 2 speaker independent connected digit recognition task [8]. Spectral domain features were used so that missing data techniques can be applied [9]. Feature vectors were obtained via a 32-channel auditory filter bank [10] distributed in frequency between 50 Hz and 3750 Hz on the ERB scale [11]. The features were supplemented with their temporal derivatives to form a 64 dimensional feature vector.

4.2. Recognition Systems

Thirteen word-level HMMs were trained on the Aurora clean speech training set. Eleven models ('1'-'9', 'oh' and 'zero') consist of 16 no-skip, left-right states with observations modelled with 7 component diagonal Gaussian mixtures. A 3-state silence model was used to model the long pauses before and after the utterance and an additional 1-state silence model was used to model the brief inter-digit pauses that may occur during long digit strings.

Four recognition systems were evaluated. The baseline system is a gender-independent 'missing data' recogniser using 'soft SNR mask' techniques described in [12]. The 'missing data' approach [9] assumes that when the speech is one of several sound sources, some spectro-temporal regions will remain uncorrupted and can be used as 'reliable evidence' for recognition. The masks of reliable evidence are derived from local SNR estimation [12]. Results with this technique are comparable to the best of those reported for systems using models trained on clean speech [12]. In the baseline system no word duration models were used. The second system is baseline + uniform word duration model (UNIWDM), i.e. all word duration penalties are set to 1 so that only the boundary can affect the search. This is to check the effect of the histogram word duration model in use. The third one is baseline + context-independent duration model (CIWDM), in which word duration penalties are calculated according to Eq. (1). The last recognition system is baseline + context-dependent duration model (CDWDM) proposed in this paper.

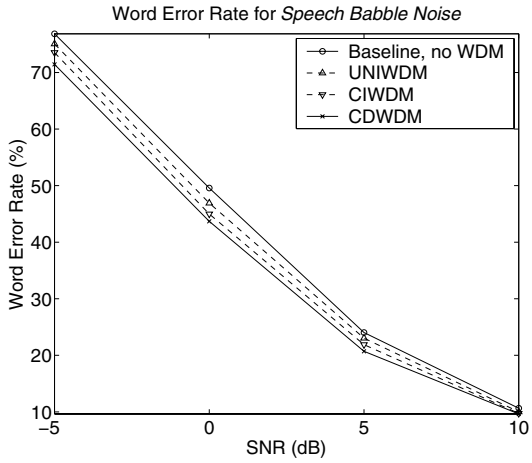

Figure 4: Word error rate for 'speech babble noise' test data at various SNR levels.

5. Results and Discussion

Recognition results for test data mixed with speech babble noise are shown in absolute word error rate (WER) in Fig. 4 and as relative improvements over the baseline system in Tab. 1. Only results of SNR levels from -5 dB to 10 dB are shown as they all converge above 10 dB and the difference is not significant. Note the UNIWDM system itself gives improvement, as mentioned in Section 3.1. The CIWDM system gives further improvement. The best performing system is that which employs context-dependent word duration modelling, especially in low SNR situations. One possible reason is that in noisy conditions missing data mask estimation becomes more difficult and therefore decoding without word duration constraints is more likely to produce word matches with inappropriate durations. This is analogous to increasing the contribution of the language model when the acoustic model is poor. Another reason is the 'missing data' systems use HMMs trained solely on clean speech; there is no re-training on noisy data. Thus in low SNR cases by introducing the duration model we can reduce the mismatch between the models and the noisy data.

To examine the word duration information encoded in the new systems, we obtained histograms from the test data by accumulating word duration examples produced in recognition, shown in Fig 5: (a) is the histogram for digit 'seven' from

Table 1: WER relative improvements over the baseline system.

	-5 dB	0 dB	5 dB	10 dB
UNIWDM	2.32%	5.39%	4.14%	5.96%
CIWDM	4.33%	9.27%	8.95%	8.52%
CDWDM	7.04%	11.89%	13.84%	9.38%

forced-alignment with the training data. (b) is the histogram from the clean test data by recognition using the baseline system. Both histograms have similar duration statistic patterns. (c) is the histogram obtained from the same test data as (a) but mixed with speech babble noise at 0 dB SNR, from recognition using the baseline system. The decoder in this case produced many word matches with very long durations (longer than 800 ms). (d-f) are from the same test data as (c) but using the UNIWDM, CIWDM and CDWDM systems, respectively. The hard boundary in UNIWDM can force the decoder to pick up word matches with durations within a range, but there are still quite a few examples at the upper boundary (d). By applying context-independent word duration penalties the histogram (e) looks more similar to (a) and with context-dependent word duration modelling the decoder produced more reasonable duration statistics (f), indicating that by introducing more realistic duration patterns better results can be achieved.

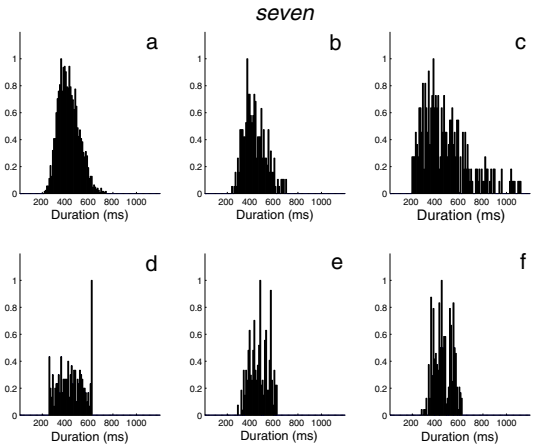

Figure 5: Word duration histograms for digit 'seven'. (a) is the histogram obtained from training data by forced-alignment. (b) is from clean test data by recognition using the baseline system. (c) is from the same test data but mixed with speech babble noise at 0 dB SNR using the baseline system. (d-f) are from the same data as (c) but using the UNIWDM, CIWDM and CDWDM systems, respectively.

6. Conclusion

We have presented techniques to explicitly model context-dependent word duration constraints. Experiments show that the approach is able to offer significant improvements over the 'missing data' recognition baseline, especially in noisy situations. However, the model assumes that the duration patterns from clean training data have approximately same statistics as those from noisy test data (the Aurora 2 test data are clean speech artificially mixed with noises). This is not always true in realistic because humans tend to slow down their speech in noise. Therefore in practical word durations need normalisation of speaking rate [4]. In future we plan to examine this and incorporate whole-word duration modelling into multi-source decoding [13].

7. Acknowledgement

This work was funded by UK EPSRC grant GR/R47400/01. Thanks to Jon Barker for discussions on decoder implementation.

8. References

[1] L. Rabiner, J. Wilpon, and F. Soong, "High performance connected digit recognition using hidden Markov models," *IEEE Trans. Acoust. Speech Signal Processing*, vol. ASSP-37, no. 8, pp. 1214–1225, Aug. 1989.

[2] K. Power, "Durational modelling for improved connected digit recognition," in *Proc. ICSLP'96*, Philadelphia, USA, 1996, pp. 885–888.

[3] M. Russell and R. Moore, "Explicit modelling of state occupancy in hidden markov models for automatic speech recognition," in *Proc. ICASSP'85*, 1985, pp. 5–8.

[4] V. Gadde, "Modeling word durations," in *Proc. ICSLP'00*, vol. 1, Beijing, China, Oct. 2000, pp. 601–604.

[5] T. Crystal, "Segmental durations in connected-speech signals: Syllabic stress," *J. Acoust. Soc. Amer.*, vol. 83, no. 4, pp. 1574–1585, 1988.

[6] D. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proc. ICASSP'92*, vol. 1, San Francisco, 1993, pp. 25–28.

[7] L. Bahl, P. Gopalakrishnan, and R. Mercer, "Search issues in large vocabulary speech recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition*, Snowbird, UT, 1993.

[8] D. Pearce and H.-G. Hirsch, "The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions," in *Proc. ICSLP'00*, vol. 4, Beijing, China, Oct. 2000, pp. 29–32.

[9] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and uncertain acoustic data," *Speech Comm.*, vol. 34, pp. 267–285, 2001.

[10] M. Cooke, "Modelling auditory processing and organisation," Ph.D. dissertation, Department of Computer Science, University of Sheffield, 1991.

[11] B. Glasberg and B. Moore, "Derivation of auditory filter shapes from notched-noise data," *Hearing Res.*, vol. 47, pp. 103–138, 1990.

[12] J. Barker, M. Cooke, and P. Green, "Robust ASR based on clean speech models: an evaluation of missing data techniques for connected digit recognition in noise," in *Proc. Eurospeech'01*, Aalborg, Denmark, 2001, pp. 213–216.

[13] J. Barker, M. Cooke, and D. Ellis, "Decoding speech in the presence of other sources," *Speech Comm.*, vol. 45, pp. 5–25, 2005.