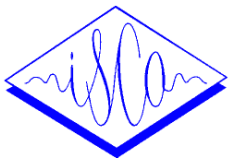

First International Conference on
Spoken Language Processing (ICSLP 90)
Kobe, Japan
November 18-22, 1990

Duration constraints for the Speech Input Interface in the MULTIWORKS Project.

Haïyan Yé* Jean Caelen

Institut de la Communication Parlée, UA 368
ENSERG/INPG-Université Stendhal 46, Av. F. Viallet F-38031 Grenoble Cedex France

ABSTRACT

We present here an early version of MULTIWORKS Speech Input Interface without syntax rules. We have shown that the duration constraints is very useful even whole-word Hidden Markov Model (HMM) is used in the case of connected word recognition.

1. INTRODUCTION

MULTIWORKS is a project for designing a MULTImedia Integrated WORKStation - an inexpensive office information system - for 1992. This machine will have several communication media : keyboard, mouse, speech I/O. MULTIWORKS is a four year ESPRIT II project (No.2105) which involves BULL, OLIVETTI, ICP/INPG, CRIN/INRIA, ... The speech I/O interface is in two parts - speech synthesis (output) and speech recognition (input). This paper concerns only the speech recognition part.

Recently, the HMM (Hidden Markov Model) approach to automatic speech recognition has become predominant approach. It can perform for continuous speech recognition (JELINEK, 1976; LEE, 1988) as well for connected or isolated word recognition (RABINER et al, 1986; GUPTA et al., 1988; IBM, 1985). The strength of HMMs lies on their probabilistic nature which enables them to extract relevant input pattern regularities from a volume of data. They have a powerful ability to automatically extract optimal parameters. All speech units and many level of knowledge can be represented within HMMs. It is for these reasons that this kind of approach has been selected for the MULTIWORKS project.

As with all other approaches, HMMs need to make some assumptions - especially structural assumptions - for constructing recognition models. With an a priori model, if a part of the model does not reflect reality, the performance of the model will be limited. This is the case with the conventional HMMs for modelling acoustic event duration, when each state of the HMM represents an acoustic event. In this kind of model the probability of occupying a particular state decreases exponentially. In fact this probability trends to have a gamma-like distribution (CRYSTAL & HOUSE, 1982). This has led FERGUSON (1980), LEVINSON (1986) to propose a variable duration model allowing explicit modelling of the probability distribution of state occupancy.

Normally if a HMM is used to represent a whole decision unit as phone or word, its state will not have a significant phonetic sense. Also its duration information should be modelled implicitly during the training phase. That means that the recognition task would be carried out correctly without direct incorporation of duration information, while this is not the case. It is repotted for isolated digit recognition with whole-word HMMs that the incorporation of duration information can improve recognition rate (HARBORG & JOHNSEN, 1988). In the experiment of RABINER et al. (1986) using whole-word HMMs for digits, the incorporation of duration information is necessary to get the same performance as that given by the DTW approach for connected-digit recognition. In this paper we try to explain this and to show the importance of duration information for the system using the whole-word HMMs and its effect on system behavior. In a connected word recognition system by concatenating word models, duration information is a useful constraints in searching for the optimal path.

Several methods have been proposed for incorporating duration information into HMMs. Among these methods, three are representative : a) variable duration HMMs (FERGUSON, 1980; LEVINSON, 1986); b) tied transition HMMs (DENG et al., 1989); and c) post-processing of duration information (RABINER et al., 1986). The first is very timing consuming and the model is very complex. The second is particularly suitable for phoneme duration modelling. The last is the most tractable of the three and needs only little more computation. The post-processing method is used in this paper.

2. THE CONSTRAINTED RECOGNITION MODEL

2.1. Basic Structure Constraints and the One-Pass Algorithm

When a speech recognition system uses whole-word HMMs for connected word recognition, three algorithms can be found to concatenate word models : the one-pass (BRIDLE et al., 1983; NEY, 1984), the level-building (MYERS & RABINER, 1981) , and the two-level (SAKOE, 1979) algorithms. The two-level algorithm and the original level-building algorithm are not left-right processes, but require several passes over the speech signal. The one-pass algorithm does not have this problem and requires less computation with respect to the above methods, but it has the difficulty of generating alternatives to the optimal decoded string. The one-pass and the level-building algorithms have been shown to be identical when syntactic knowledge is introduced (GODIN & LOCKWOOD, 1989) The one-pass algorithm is chosen as the approach for our project.

Initially, the above three algorithms were proposed for a dynamic programming based technique in which reference templates were used. As there is no the notion of state in the template, there is no state duration modelling. The duration of phonetic events is explicitly represented by the duration corresponding with the template. RABINER et al. (1986) have implemented the level-building algorithm with a HMM based technique. In this section we describe the implementation of the one-pass algorithm to concatenate HMMs.

The basic idea of the one-pass algorithm is to treat **all** the reference model candidates time-synchronously : each input frame is compared with all the references before the next input frame. In Fig.1, this principal is illustrated : there are three HMMs : M1, M2, M3; the optimal path is traced in bold, which produces the decoded string M2 M1 M3; the two vectors pM, pF allow backtracking.

We give below the flow-diagram of the one-pass algorithm.

Flow-diagram of the OP algorithm

INITIALIZATION
LOOP ON INPUT FRAMES t = 0 .. N-1
LOOP ON REFERENCE MODELS k = 0 .. K-1
LOOP ON STATES j = 0 .. NoEtat-1 of MODEL k
test condition for changing model
Compute the column of cumulative probabilities and the column of backpointers
Update from-model and from-frame arrays
Store the last columns of probabilities and backpointers for computation at time i+1
BACKTRACKING OF THE OPTIMAL PATH

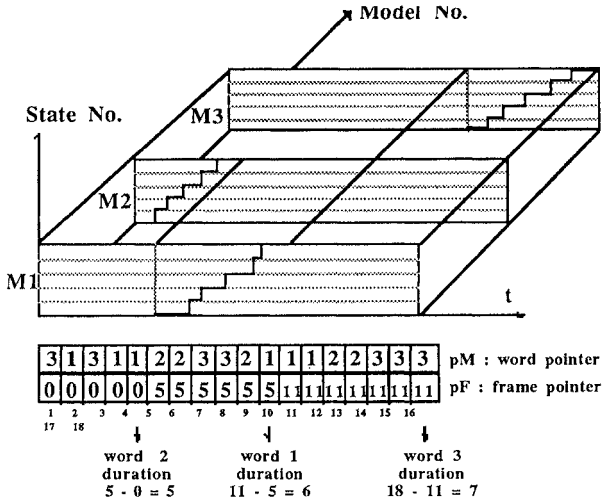

Fig.1 The principal of the one-pass algorithm with HMMs.

The most important step in this algorithm is the between model test : whether we should change the model when we progress from frame t-1 to frame t. This test can be illustrated by the following figure :

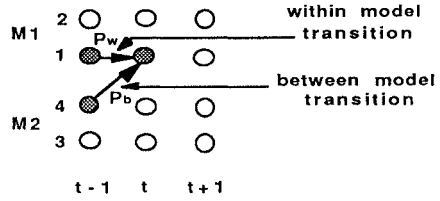

Fig.2 The rule for concatenating HMMs used by one-pass algorithm : if (Pw > Pb) then within-model transition else between-model transition.

We can see that this rule is very straightforward and artificial. The direct application of this algorithm without any other constraints can only provide very poor results. We will discuss this point in section 3.

2.2. Duration constraints

Incorporation of duration information is shown to be very interesting for either whole-word HMMs or phonetic based HMMs (Fig.3). Three kinds of duration information can be defined : 1) Phonetic unit duration, 2) HMM state duration, 2) HMM state timing. We give a brief definition of each of these duration below.

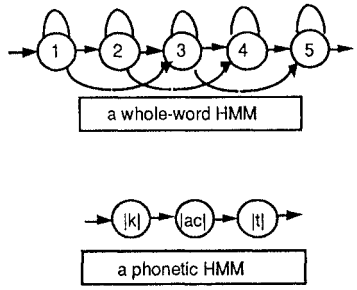

Fig.3 Examples of whole-word HMMs (at the top) and phonetic based HMMs (at the bottom). Whole-word HMMs have the same number of states (so the same structure) for all the vocabulary, for example 5 states here, while the number of state for phonetic based HMMs depends on the number of decision units in the word, for example 3 states here for word "c-a-t".

In this paper, duration information is studied with whole-word HMMs. Each HMM unit represents a word and each state has no explicit phonetic sense.

A) Phonetic unit duration (HMM duration)

This measure is the duration of a phonetic unit that is modelled by a HMM. The original HMMs have no constraints on this duration. In our case this unit is a word, so this duration is also the HMM duration. We do not know the distribution of word duration for continuous speech. RABINER et al. (1989) make a Gaussian assumption for this distribution in their high performance connected-digit recognition system. This assumption is retained in this paper. Fig.4 shows the distributions of two French word durations pronounced by a male speaker with its mean and variance. The following histograms were built with 50 utterances for each word.

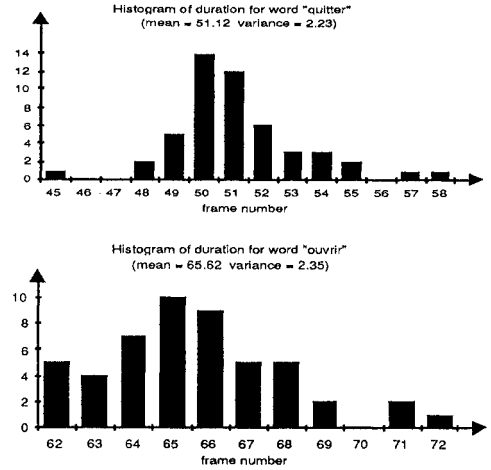

Fig.4 Distributions of two French word durations, quitter (quit) and ouvrir (open), pronounced by a male speaker with its mean and variance.

B) State duration

HMM state duration can be expressed by the measure of l/L where **L** is total frame number of the word and **l** is the frame number staying in a state. There is a l/L measure for each state. l/L can be considered as a normalized time which is always less than or equal to 1. In order to build HMM state duration histograms, the following procedure is used : after a word HMM has been trained with 50 utterances, each utterance will be rewarped to the same HMM by the Viterbi algorithm. By backtracking for the optimal path, the HMM state durations for an utterance can found. After that these durations can be quantified in to the histogram as the following figure.

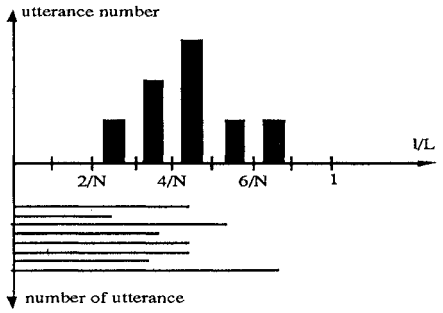

Fig. 5 Quantifying HMM state duration into a histogram. The x-axis represents normalized time (l/L) which is quantized into N intervals, the negative y-axis represents the number of the utterance, the positive y-axis represents the number of utterances accumulated in the corresponding interval.

In Fig.5, an example of a histogram for one state of a HMM is presented. The x-axis represents normalized time (l/L) which is quantized into N intervals, the negative y-axis represents the number of the utterance, the positive y-axis represents the number of utterances accumulated in the corresponding interval. The state duration histograms for "quitter" are given below (with 50 utterances). These histograms can be modelled by a Gaussian distribution.

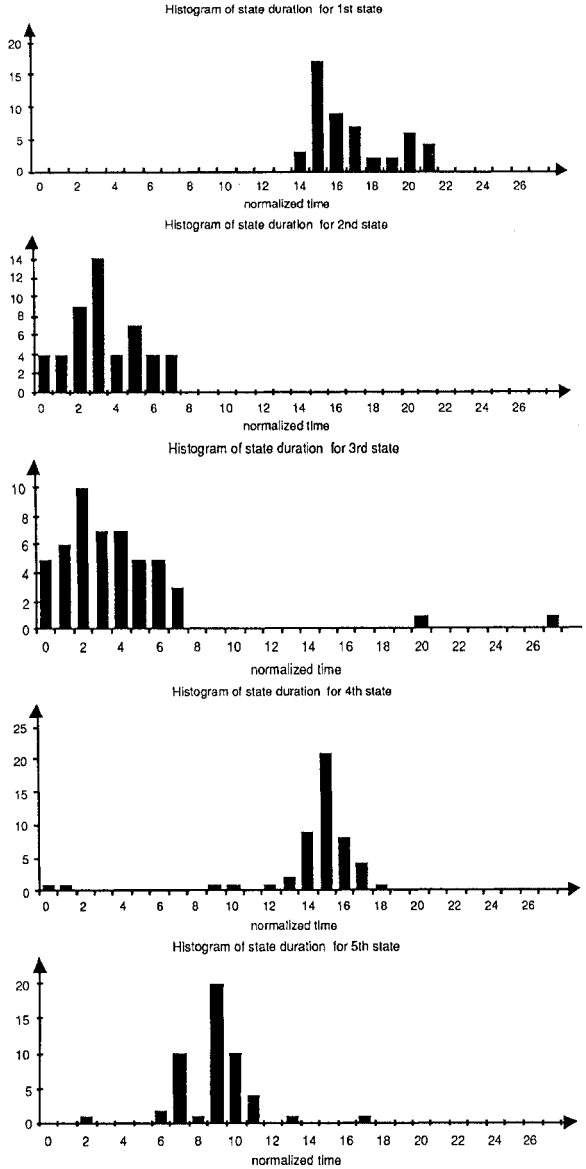

Fig.6 The HMM state duration histograms for word "quitter". HMM has 5 states. Histograms are built from 50 utterances. The dimension of the x-axis is the index Q where Q=50*(normalized time). Their Gaussian parameters (mean and standard derivation) are,
for the 1st state : mean=0.344095; σ=0.042520;
for the 2nd state : mean=0.080170; σ=0.037422;
for the 3rd state : mean=0.093101; σ =0.090017;
for the 4th state : mean=0.295262; σ=0.063972;
for the 5th state : mean=0.187372; σ=0.039757.

C) State timing

HMM state duration only takes into account the time spent in each state. It does not give the specific corresponding time. A measure can be defined to supply this information as : **i/L** where **L** is total frame number for the word and **i** the number of the last frame staying in a state. The same procedure as for the last section is used for building a histogram. The HMM state timing histograms for the word "quitter" are given in Fig.7.

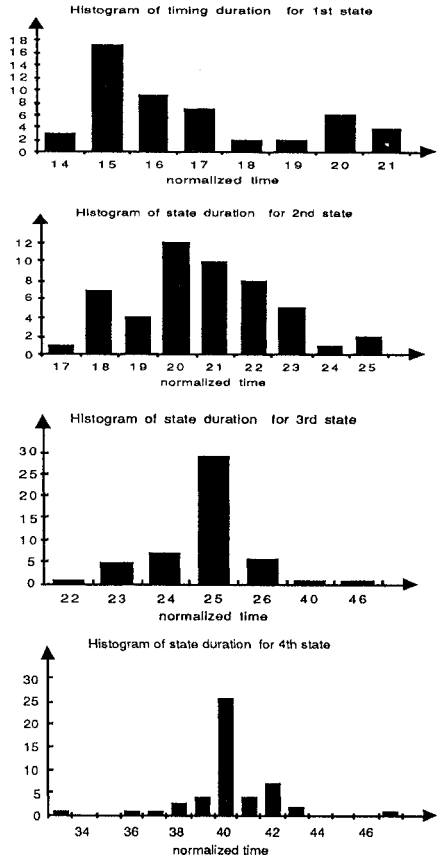

Fig.7 HMM state timing histograms for the words "quitter". The HMM has 5 states. Histograms are built with 50 utterances. Their Gaussian parameters are,
for the 1st state : mean=0.344095; σ=0.042520;
for the 2nd state : mean= 0.424265; σ= 0.035645 ;
for the 3rd state : mean= 0.517366; σ=0.074762;
for the 4th state : mean= 0.812628; σ= 0.03975;
for the 5th state : mean=1; σ=0.

D) Incorporation of duration constraints

All three above duration measures can be modelled by the Gaussian distribution N(x,μ,σ) where μ and σ are the mean and variance respectively. The above histograms show that this assumption is not so far from reality. A more detailed statistical study may be needed to confirm this assumption in the future.

The HMM duration parameters are computed during the HMM training phase. After this training phase input utterances are rewarped to the trained HMM in order to find the mean and σ of HMM state and timing duration. These parameters will be stored in a reference file for recognition.

We propose here several ways for incorporating duration information into the recognition algorithm. All of these can be considered as constraints on the between-models transition condition.

i) word duration

This is what we have named the HMM duration or phonetic unit duration. This duration information can act at two levels : global and local.

Global constraint : when the actual duration staying in a HMM during the optimal path search is less than to half of mean duration of the same HMM, the between-model transition is not allowed :

if $(t2-t1) < m_m/2$ then between-model transition is forbidden

where t1 is the frame number entering the HMM, t2 is actual frame number and m_m is mean duration of the corresponding HMM.

Local constraint : the probability $P_m = N_m(t, m_m, \sigma_m)$ will contribute to the cumulated probability when the one-pass algorithm changes HMMs during the best path search :

$$\log(P_c) = \log(P_c) + W_m * \log(P_m)$$

ii) HMM state duration and state timing

The state duration constraint is evaluated as $P_S = N_S(i/L, m_S, \sigma_S)$. Sate timing is computed in a similar way : $P_t = N_t(i/L, m_t, \sigma_t)$. These will be added to the cumulated probability when the one-pass algorithm changes HMMs during the best path search. These two conditions are very important for constraining the warping algorithm to find the best path.

$$\log(P_c) = \log(P_c) + W_S * \log(P_S) + W_t * \log(P_t)$$

iii) constraint for quitting a HMM

When a path leaves a model, some conditions can be imposed, for example, the probability of the last state must be greater than to that any of other states or Pr(state N) > Pr(state N-1) > ... > Pr(state 1). This constraint will favour good word candidates in the decoded string, because they will easily satisfy this constraint.

3. EVALUATION

The one-pass algorithm and above duration constraints were implemented and evaluated. The evaluation was carried out with ten editor command words such as ouvrir (open), quitter (quit), lister (list). The whole-word HMMs are built with isolated utterances (50 repetitions for each word) during the training phase. With this training data, the Viterbi or forward algorithm, i.e. isolated word recognition, give 100% recognition rate for the first choice.

Once the HMMs were built, two experiments were carried out to evaluate the recognition performance of the one-pass algorithm: with connected words and with isolated words.

3.1 Experiment 1

Ten connected words were used. There are two words in each connected word (our experiences show that 3-words connected word is easier to recognize than 2-words connected word). Each connected word was pronounced ten times. The recognition performances as function of Wm (word duration weighting), Ws (state duration weighting) and Wt (timing duration weighting) are listed in the following table.

Wm	Ws	Wt	Recognition rate
0.0	0.0	0.00	50.0%
0.6	0.0	0.00	82.0%
0.0	1.0	0.00	83.0%
0.0	0.0	0.02	56.0%
0.6	1.0	0.02	93.0%
0.6	1.0	0.05	93.0%

3.2 Experiment 2

The second experiment consisted of recognizing isolated words with the one-pass algorithm. 500 utterances (50 repetitions * 10 words) for HMM training are used. The recognition performance table is listed below :

Wm	Ws	Wt	Recognition rate
0.0	0.0	0.00	97.6%
0.0	0.0	0.01	98.6%
0.6	0.0	0.00	100.0%
0.0	1.0	0.00	100.0%
0.6	1.0	0.01	100.0%

4. DISCUSSIONS AND CONCLUSION

If we examine the state sequence of a HMM in detail, we find that the duration of occupying a state and the moment of leaving a state are very important for constraining the Viterbi algorithm to stay near to the best path. This is shown clearly by the above evaluation. If there are no duration constraints, the one-pass algorithm tends to split input signals into a lot of small pieces because Pb is often less strongly penalized than Pw (Fig.2). Error analysis shows that insertion is the most important error source for the one-pass algorithm. Thus a dynamic penalty is necessary for transition Pb. From the above tables, we see that the recognition performance is drastically improved. The word duration and the state duration are the two important factors for improving recognition rate. This improvement is obtained by constraining the original one-pass algorithm. By using these constraints, word duration, ignored by the whole-word HMM, is explicitly modelled and the state duration, even though it does not have any phonetic sense, can give some information to show how good the input signal is matched with a HMM. The constraints introduced by state duration permit the model to reject bad paths during between-model transition. The evaluation is carried out with HMMs trained with isolated utterances. Better training algorithm can be used with connected words data basis. In this work, we have shown that stationary speech compression can also improve speech recognition. But we have not time to develop it in this paper. We are working now towards introducing the word coarticulation constraints and high level constraints into this model.

REFERENCES

BRIDLE J.S., BROWN M.D. & CHAMBERLAIN R.M. (1983)
The Radio and Electronic Engineer 53, 167-175.

CRYSTAL T.H. & HOUSE A.S. (1982)
J. Acoust. Soc. Am. 72, 705-716.

DENG L., LENNIG M. & MERMELSTEIN P. (1989)
J. Acoust. Soc. Am. 86, 540-548.

FERGUSON J.D. (1980)
Proceeding of the Symposium on the Application of Hidden Markov Models to Text and Speech. (J.D. FERGUSON, ed.) 143-179.

GODIN C. & LOCKWOOD P. (1989)
Computer Speech and Language 3, 169-198.

GUPTA V., L., LENNIG M. & MERMELSTEIN P. (1988)
J. Acoust. Soc. Am. 84, 2007-2017.

HARBORG E. & JOHNSEN M.H. (1988)
EURASIP, Signal Processing IV: Theories and Applications Eds. J.L. Lacoume, A. Chehikian, N. Martin, and J. Malbos Elsevier Science Publishers B.V. (North-Holland).

IBM speech recognition group. (1985)
ICASSP March, 1985.

JELINEK F. (1976)
Proc. IEEE, Vol. 64, No.4, 532-556.

LEE K.F. (1988)
Carnegie Mellon, CMU-CS-88-148.

LEVINSON S.E. (1986)
Computer Speech and Language 1, 29-45.

MYERS C. & RABINER L.R. (1981)
IEEE Trans. ASSP-29, 284-297.

NEY H. (1984)
IEEE Trans. ASSP-32, 263-271.

RABINER L.R., WILPON J.G. & JUANG B.H. (1986)
Computer speech and language 1, 167-197.

RABINER L.R., WILPON J.G. & SOONG F. K. (1989)
IEEE Trans. ASSP-37, 1214-1225.

SAKOE H. (1979)
IEEE Trans. ASSP-27, 588-595.

* Now with Ascom Tech AG, Ziegelmattstr. 1-15, CH-4503, Swistzerland