Using segment probabilities for speech recognition tasks[1]

Luis Javier Rodríguez Fuentes, Inés Torres Barañano, Amparo Varona Fernández
Departamento de Electricidad y Electrónica. Facultad de Ciencias. Universidad del País Vasco.
Apartado 644. 48080 Bilbao. SPAIN.
e-mail: luisja@we.lc.ehu.es

Abstract

Our experience with an automatic dialog system providing train timetable information shows that speakers tend to use very long spontaneous utterances, specially in the first turn. The utterances might be composed of three or more sentences, reaching a duration of about 30 seconds or even more. This could lead to very low or even null acoustic probabilities when recognizing the whole utterance.

To overcome this problem two different strategies might be considered. First, the acoustic probability computation could be re-initialized by identifying the points where each component sentence finish, thus recognizing individual sentences inside each turn. This could be achieved by incorporating specific "sentence end" probabilities to the language model. On the other hand, the acoustic probability computation procedure itself could be modified, in two possible ways: a) by normalizing the acoustic probabilities at each frame, or b) by restricting the computation to a short segment, instead of the whole signal from the beginning.

In this paper we explore the effect of applying this latter approach. The recognizing procedure was slightly modified by adding an auxiliary sequence of probabilities at each trellis node. Both memory requirements and time complexity were incremented, because each sequence entry had to be updated at each frame. Preliminary acoustic-phonetic decoding experiments were carried out, using the same acoustic models (multiple codebook discrete HMMs) and different segment lengths to compute the acoustic probabilities. A phonetically balanced spanish database containing read speech sampled at 16 kHz was used, with 1529 sentences for training and 493 independent sentences for testing purposes. The same database was filtered and resampled at 8 kHz, to simulate the conditions on which our spontaneous speech database was acquired, so that the same experiments were reproduced at 8 kHz. Results show that beyond a certain segment length (around 20 frames) performance converges to the optimal. This suggests that duration-independent speech recognition could be performed by considering consecutive-overlapping segment probabilities.

1. Introduction.

Our experience with a spontaneous speech database, obtained from an automatic dialog system providing train timetable information, shows that speakers tend to use very long utterances, specially in the first turn. Our current recognizer, which uses discrete hidden Markov models as acoustic models, combines emission and transition probabilities to update at each trellis node the acoustic probability, being this combined with the language model probability whenever a word transition takes place. As the utterance is scanned, these probability values decrease monotonously, so that some of them could reach a null value. In fact, some suboptimal paths, which eventually could be part of the optimal path, might be removed from the trellis. Moreover, *all* of the probabilities could reach a null value, thus breaking the recognition procedure, and no hipothesis could be made.

To cope with this problem we considered two different strategies. The first one, which will not be covered in this paper, would introduce special “sentence end” probabilities in the language model. These special probabilities could be easily trained using a representative subset of turns. The probability computation would be re-initialized each time a sentence transition happened at any trellis node. Finally, an optimal utterance segmentation into various individual sentences would be obtained. Each sentence would give its own probability score. The probabilities of these sentences would be stored during search and combined at the end to obtain the probability for the whole utterance. By the way, this segmentation might help higher processing modules to distinguish different meanings or purposes (speech acts) inside a turn. However, a very long sentence could be found yet and the issue of probability cancellation would not be overcome.

[1] Work partially supported by the spanish CICYT, under proyect TIC98-0423-C06-03.

The second approach would focus on *improving* the acoustic probability computation itself, by avoiding the monotonous decrease of the values stored at the trellis nodes. This could be achieved either by normalizing the acoustic probabilities at each frame, or by restricting the probability computation to a short time interval. We chose this latter approach because it could be suitable for a confidence measure based on the likelihood of the speech segment being analysed. However, with the aim of comparing their performance, the normalizing approach was implemented and tested too.

Section 2 briefly describes the properties of our spontaneous speech database, particularly the length of the user utterances, which motivated this work. Section 3 outlines the two methods used to overcome the issue of probability cancellation. Preliminary experimentation was carried out using a phonetically balanced read speech database. This database along with the main features of our acoustic-phonetic decoder are described in Section 4. Both recognition rates and time costs are showed and discussed in Section 5. Finally, Section 6 gives some conclusions and directions for future research.

2. Motivation.

Although the problem addressed in this work is common both to read and spontaneous speech, we only faced it recently when we moved from read to spontaneous speech in the context of a interactive dialog system providing prices and timetables for long distance train travels [1]. Before that, our off-line experiments dealt with read speech signals of about five seconds, and our on-line speech recognition prototype could handle signals lasting up to ten seconds [2]. This was enough since we were recognizing simple –not interactive– queries to access a geographical database. So we never found the event of probability cancellation.

Our spontaneous speech database consists of 227 dialogues, recorded across telephone lines, between instructed users and a Wizard of Oz mechanism which elaborated answers according to a previously designed grammar [3]. Users were told to use short sentences but they were free to interact. In fact, they tended to use very long questions, including complex explanations. The recording system posed no limit on signal length. Without counting the empty turns which took place when the user waited for the machine to synthetize the answer, the database includes 1657 user turns, which last 150 minutes approximately. Certainly this is not enough to draw up strong conclusions, but it is to study spontaneous speech features and to carry out preliminary experimentation which will pave the way for future research.

Table I. Statistics showing the average, standard deviation and maximum duration for the first twenty user turns of the 227 dialogues.

Turn	Average duration (sec)	Standard deviation (sec)	Maximum duration (sec)
U0	10.939292	5.561586	34.106450
U1	4.396883	4.258852	23.422000
U2	4.573519	4.123588	22.737600
U3	4.848704	3.657449	20.114200
U4	4.292786	3.406280	18.507900
U5	4.114686	3.164122	16.732800
U6	4.143523	3.515365	18.441100
U7	4.297872	4.067183	29.953000
U8	4.223021	3.640354	16.711000
U9	4.894576	6.312496	50.312000
U10	5.186244	7.469937	48.272000
U11	3.985444	3.849036	23.270000
U12	4.505000	4.193358	21.483000
U13	4.779758	4.263118	20.532000
U14	3.795720	2.865028	9.240000
U15	3.976700	2.755065	9.810000
U16	6.339667	3.873562	15.184000
U17	3.848200	3.213877	12.851000
U18	4.993571	5.396069	19.679000
U19	2.547875	1.904310	6.806000
U20	4.763167	3.565036	10.576000

To enlighten how long the turns could be, we show in Table I some statistics about the first twenty turns: average duration, standard deviation and maximum value. Clearly the first turn average is more than two times the others. But this does not mean very much, because we found peak durations in turns 9 (50.31 seconds) and 10 (48.27 seconds). Most of the averages lie between 4 and 5 seconds, because many times users answered with simple "yes" or "no" monosyllables, wich compensate for the long answers. These monosyllables were uniformly distributed among all turns but the first. So we can conclude that long turns could appear at any time. Although probability cancellation might not always happen, it seems necessary to modify the recognition algorithm so that probabilities do not become null.

3. Proposed changes to the recognition algorithm.

This Section will focus on the acoustic component of the recognizer. The language model would just contribute with additional probabilities at each word transition, thus accelerating the pace at which accumulated probability decreases; the search would be restricted by lexical baseforms, but the reasoning would be the same. We will assume that hidden Markov models are used as acoustic models.

The key ingredient of the procedure used to search the best sentence hypothesis, widely known as Viterbi algorithm [4], is the optimization step, which could be formulated as follows: for every state q at time t, search the state p^* at time $t-1$ that maximizes the accumulated probability, according to the following expression:

$$p^*(q,t) = \arg\max_{\forall p}\left\{\Pr(p,t-1)\cdot a(p,q)\cdot b(o_t,q)\right\}$$

where $Pr(p,t-1)$ is the accumulated probability –previously maximized– at state p and time $t-1$, $a(p,q)$ is the transition probability from state p to state q, which is assumed to be independent of t, o_t is the acoustic observation at time t, and $b(o_t,q)$ is the probability of emitting the observation o_t at state q. Note that this latter probability does not depend on p, so it is usually left aside. Once optimized, the accumulated probability at state q and time t is assigned the quantity between braces:

$$\Pr(q,t) = \Pr(p^*,t-1)\cdot a(p^*,q)\cdot b(o_t,q)$$

and the state p^* that maximizes $Pr(q,t)$, stored in $R[q,t]$, a matrix that will allow us to recover the optimal path. The time complexity of this procedure is lower than expected. Only a few states are explored to determine the optimal $p^*(q,t)$, because most state transitions are bound to happen inside a HMM, and only transitions between HMMs –or between lexical units when using a language model– can reach a sizeable amount of states. Therefore, calling Q to the total number of states and T to the length of the sequence of observations, time complexity will be closer to $\Omega(QT)$ than to $O(Q^2T)$. On the other hand, memory requirements are given by the matrix $R[q,t]$ mentioned above, so space complexity will be exactly $\Theta(QT)$.

3.1. Segment probabilities.

We define *segment probability* $Pr_M(q,t)$ as the accumulated probability corresponding to the optimal path beginning at time $t-M$ and finishing at state q at time t, being M the segment length. This value could be computed as follows:

$$\Pr_M(q,t) = b\left(o_{t-M}, p_{q,t,M}(t-M)\right)\prod_{m=M-1}^{0} a\left(p_{q,t,M}(t-m-1), p_{q,t,M}(t-m)\right)\cdot b\left(o_{t-m}, p_{q,t,M}(t-m)\right)$$

where $p_{q,t,M}(t-k)$ stands for the state occupied at time $t-k$ in the optimal path of length M that finishes at state q at time t. We also define the auxiliary values:

$$\Pr_M(q,t,n) = b\left(o_{t-n}, p_{q,t,M}(t-n)\right)\prod_{m=n-1}^{0} a\left(p_{q,t,M}(t-m-1), p_{q,t,M}(t-m)\right)\cdot b\left(o_{t-m}, p_{q,t,M}(t-m)\right)$$

which represent the accumulated probability from time $t-n$ to time t in the optimal path of length M that finishes at state q at time t, with $n = 0,1,2,\ldots,M-1$. Note that $Pr_M(q,t,0) = b(o_t,q)$.

Given the preceding definitions, a procedure can be written to update at each time t both the segment probability $Pr_M(q,t)$ and the auxiliary values $Pr_M(q,t,n)$. Firstly the optimization principle must be applied: for every state q at time t, search the state p_M^* at time $t-1$ that maximizes the segment probability, according to the following expression:

$$p^*_M(q,t) = \arg\max_{\forall p}\left\{\Pr_M(p,t-1,M-1)\cdot a(p,q)\cdot b(o_t,q)\right\}$$

Then the segment probability and the auxiliary values can be updated:

$$\Pr_M(q,t) = \Pr_M\left(p^*_M,t-1,M-1\right)\cdot a\left(p^*_M,q\right)\cdot b(o_t,q)$$

$$\Pr_M(q,t,n) = \Pr_M\left(p^*_M,t-1,n-1\right)\cdot a\left(p^*_M,q\right)\cdot b(o_t,q) \quad n = 1,2,...,M-1$$

$$\Pr_M(q,t,0) = b(o_t,q)$$

At first sight, the resulting algorithm requires a new array of M auxiliary values for each segment probability $Pr_M(q,t)$, implying a space complexity of $\Theta(QTM)$. But efficient implementations reduce the matrix $Pr_M(q,t)$ to only two columns, standing for present time probabilities and previous time probabilities, thus resulting a space complexity of $\Theta(QM)$. Since usually $T>>M$, it follows that the matrix $R[q,t]$ still dominates the space complexity, which remains $\Theta(QT)$. On the other hand, the cycle for updating the auxiliary values $Pr_M(q,t,n)$, with $n = 0, 1, 2, ..., M-1$, makes the time complexity increase linearly with M, which gives a value between $\Omega(QTM)$ and $O(Q^2TM)$.

3.2. Normalized probabilities.

This second approach is remarkably simple and efficient. Firstly the maximum value of the accumulated probabilities is updated during the optimization cycle at each time t. We will refer to it as $MaxPr(t)$. Then each value $Pr(q,t)$ is divided by $MaxPr(t)$:

$$Max\Pr(t) = \max_{\forall q}\left\{\Pr(q,t)\right\}$$

$$\forall q \quad \Pr(q,t) \leftarrow \Pr(q,t) / Max\Pr(t)$$

The maximization instruction slightly raises the hidden constant corresponding to the optimization cycle. The updating cycle traverses just the same values but with a much smaller hidden constant than that of the optimization cycle. So a little increase of the time cost must be expected, whereas the time complexity remains between $\Omega(QT)$ and $O(Q^2T)$. Obviously, the space complexity is not increased at all, because only one additional real value is neccesary to store the maximum probability at each time.

It could be discussed the effect of such a normalization, but since all the accumulated probabilities are divided by the same value, we conclude that the optimization procedure will produce the same results than it would without normalization. At the end, the optimal path will give probability 1.00. The accumulated probability corresponding to this path could be obtained by multiplying the maxima, as follows:

$$\Pr(best\ \ path) = \prod_{t=1}^{T} Max\Pr(t)$$

4. Experimental framework.

Our aim was to implement and to verify the performance of the two alternative recognition algorithms proposed in Section 3. A phonetically balanced read speech database in Spanish was used for preliminary experimentation, because the spontaneous speech database which motivated this work was not completely arranged at the time of writing this paper. The database contained 1529 sentences for training, involving around 60,000 phone samples. For testing purposes a speaker independent corpus composed of 493 sentences was used. This database was originally acquired at 16 kHz, but our target spontaneous speech database was acquired across telephone lines at 8 kHz. So the read speech database was filtered and re-sampled at 8 kHz to simulate the signal conditions of the spontaneous speech database.

Signal analysis was made as described in [2], following conventions of the Entropic Hidden Markov Modeling Toolkit (HTK) [5], being the frame length 25 milliseconds and the interframe distance 10 milliseconds (100 frames per second). Acoustic parameters at 16 kHz included 12 filter bank mel-scale cepstral coefficients, plus their first and second derivatives, energy and its first derivative. Signal analysis was made the same way at 8 kHz, using only 10 cepstral coefficients. Both at 8 and 16 kHz, the standard LBG vector quantization procedure was applied to obtain four codebooks, each containing 256 centroids, corresponding to cepstral coefficients, their first and second derivatives, and a 2-component vector composed of energy and its first derivative.

Discrete left-to-right hidden Markov models, with three looped states and four discrete observation distributions per state –corresponding to the four mentioned vector quantization codebooks–, were used as acoustic models. HMM parameters were estimated using Baum-Welch and forced Viterbi procedures, applying the maximum likelihood criterion.

An acoustic-phonetic decoding task was posed as benchmark, so neither lexical baseforms nor language model were necessary. Although it would help recognition, no phonological model was applied. A set of 23 context-independent phone-like units, plus one special unit for silence, disposed as usual in parallel, were used for recognition, applying a transition weight between models of 1/24.

5. Results.

Experiments were carried out for 16 kHz and 8 kHz. Just one set of discrete HMMs was trained and then used for recognition in each case. As a reference, the standard Viterbi recognition procedure was applied. Then an experiment applying the normalized version proposed in Section 3.2 was carried out. Finally a series of ten experiments for different segment lengths (1, 2, 3, 4, 5, 10, 15, 20, 25 and 30 frames), applying the approach proposed in Section 3.1, was carried out. Recognition rates and time costs are shown in Table II (16 kHz) and Table III (8 kHz).

Table II. Recognition rates and time costs for one set of HMMs, trained with a 16 kHz database, and three different recognition algorithms.

Recognition procedure		**% Recognition**	**Time spent (sec)**
Viterbi		61.705759	85.014058
Normalized Viterbi		61.705759	89.421321
Segmental Viterbi	**Segment length**		
	1	38.199627	87.606550
	2	48.245580	95.522234
	3	55.229478	100.269432
	4	58.675140	108.944843
	5	60.119516	113.766735
	10	61.546085	140.394050
	15	61.665080	169.382801
	20	61.703901	194.510452
	25	61.707278	221.662240
	30	61.705759	247.815799

Table III. Recognition rates and time costs for one set of HMMs, trained with a 8 kHz database, and three different recognition algorithms.

Recognition procedure		**% Recognition**	**Time spent (sec)**
Viterbi		56.069045	86.212461
Normalized Viterbi		56.069045	89.016965
Segmental Viterbi	**Segment length**		
	1	28.759592	87.769069
	2	33.986978	94.815188
	3	44.279709	100.437729
	4	49.631159	108.299926
	5	52.392829	113.875988
	10	55.863321	140.682120
	15	56.041222	170.302838
	20	56.058930	193.514446
	25	56.055937	221.808531
	30	56.069045	247.411304

For a lack of space we do not include graphical representations of these two series of experiments. However, it can be easily observed that the normalized Viterbi outperformed the approach based on segment probabilities in terms of time costs. On the other hand, this latter version converged to an optimal performance as the segment length was increased. A coherent behaviour can be observed in both series of experiments, because optimal performance was reached with a segment length of about 20 frames (0.2 seconds). It is surprising the relatively high performance attained with a segment length of only five frames.

Robustness against probability cancellation was more directly and more efficiently achieved with the normalized Viterbi procedure. Note that it spent only 5% more time than the reference implementation, while the version based on segment probabilities achieved the same performance in more than twice that time. These results confirm the discussion about time complexity showed in Section 3. However, as said above, the approach based on segment probabilities could be applied to compute in a straightforward way some kind of confidence measure, thus allowing to monitor how the recognition progresses, which is crucial for on-line systems.

6. Conclusions and future research.

The issue of probability cancellation due to very long input signals was addressed in this paper. We opted for approaches based on the acoustic component, specifically for two variations to the standard Viterbi algorithm, being the first a simple normalization, and the second a more complex procedure which used segment probabilities. It was found that using probabilities corresponding to segments of about 20 frames gave almost the same performance than using probabilities corresponding to the whole signal. However, using normalized probabilities achieved the same goal with a much smaller time cost.

Future work will include to repeat these experiments over our spontaneous speech database, where probability cancellation should really happen. As said above, we will try to define a confidence measure based on segment probabilities. Also the use of "sentence end" probabilities in the language model will be explored, because though not useful for avoiding probability cancellation, it could be used to segment dialogue turns into individual sentences, which would considerably help the speech understanding and dialog management modules.

7. References.

[1] A. Bonafonte. "Desarrollo de un sistema de diálogo para habla espontánea en un dominio semántico restringido". *Documento interno: memoria. Proyecto TIC98-0423-C06. Junio 1998*.

[2] L.J. Rodriguez, I. Torres, J.M. Alcaide, A. Varona, K. López de Ipiña, M. Peñagarikano, G. Bordel. "An integrated system for Spanish CSR tasks". *Proceedings of EUROSPEECH-99, Vol. 2, pp. 951-954.*

[3] I. Esquerra, A. Sesma, J.B. Mariño. "Generación de respuesta para el Mago de Oz". *Documento interno: BS61AV23. Proyecto TIC98-0423-C06. Diciembre 1999.*

[4] L.R. Rabiner. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition". *Proceedings of the IEEE, Vol. 77, N. 2, pp. 257-286. February 1989.*

[5] S. Young, J. Odell, D. Ollason, V. Valtchev, P. Woodland. "The HTK Book. Hidden Markov Model Toolkit v. 2.1". *Entropic Cambridge Research Laboratory. March 1997*.