TRAINING OF ACROSS-WORD PHONEME MODELS FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

Achim Sixtus and Hermann Ney

Lehrstuhl für Informatik VI, RWTH Aachen
University of Technology, 52056 Aachen, Germany
{sixtus,ney}@informatik.rwth-aachen.de

ABSTRACT

Today's speech recognition systems use across-word context dependent phoneme models to capture coarticulation across word boundaries. While there are several publications about the organization of across-word model search, there are hardly any descriptions about the training of across-word models.

In this paper, we discuss in detail how well performing across-word models can be trained. Starting from a conventional within-word model training we propose two training methods for across-word models and compare them with respect to computational effort and recognition performance. Then, we discuss how the across-word training should be parameterized. The experiments show that in comparison with within-word models across-word models require a substantially higher number of tied states for optimal recognition performance. Furthermore, we show that position dependent triphones outperform position independent triphones significantly.

We present results on the 1994 20k North American Business (NAB) task and on the 1997 Hub4 Broadcast News task.

1. INTRODUCTION

It is well known that the recognition accuracy can be improved significantly by considering context dependency of the phonemes not only within the words but also across word boundaries. While there are several publications about the organization of across-word model search (eg. [1], [2], [3]), the *training* of across-word phoneme models is studied only in very early publications (e.g. [4]).

In this paper we discuss in detail the modifications to the RWTH within-word model training method described in [5] to enable across-word model training. The paper is organized as follows:

- First, we discuss the modifications applied to the HMM automata of the training utterances if across-word models instead of within-word models are used.
- Next, we propose two training methods which take these modifications into account. We compare these methods with regard to recognition accuracy and computational effort.
- Finally, we discuss how the parametrization of the training should be modified if across-word models instead of within-word models are to be trained. In particular we study
  - the number of tied states needed for good recognition performance, and
  - the effect of position dependent triphones.

We will report on experimental results on the 1994 20k NAB development set and on the 1997 evaluation set of Hub4 Broadcast News.

2. EXPERIMENTAL SETUP

The experiments to be reported in this paper were performed with our single-pass across-word trigram recognizer. More details about this system are given in [6]. Results are presented on the NAB'94 H1 development test set and on the 1997 evaluation test set of Hub4 broadcast news. In the following, these corpora are called "NAB20k" and "HUB4". Statistics about the training and test set are given in Tab. 1. On HUB4 the numbers of speakers are only estimates since not for all utterances speaker identities are given.

Table 1. Statistics of the corpora considered in this paper

	NAB20k		HUB4	
	training WSJ0+1	test dev94 H1	training '96 '97	test eval '97
amount of ac. data	81.4h	0.8h	96.5h	2.9h
silence portion	26%	18%	14%	12%
# speaker	284	20	∼3157	∼116
# utterances	37,571	310	26,167	657
# running words	649,624	7,387	1,053,052	32,834
# running phonemes	2,691,352	-	3,927,338	-
perpl. (Trigram LM)	-	124.5	-	187.5
words in lexicon	-	19,977	-	65,491
# lex. entries	-	22,412	-	74,656

The acoustic vectors are computed by applying a linear discriminant analysis on several adjacent vectors consisting of 16 cepstral coefficients without derivatives.

The acoustic model consists of triphones which are represented by 6-state HMMs with skip, forward and loop transitions. Gaussian mixtures with a globally pooled diagonal covariance are used for modelling the HMM states which are tied using a decision tree. Silence is modelled using a single state HMM which is separated from the states of the other HMMs and not included in state tying. During training the Viterbi approximation is applied.

The NAB system uses gender independent models, the HUB4 system uses gender dependent models. If not stated differently 7001 tied states were used in the NAB system and 4001 tied states for each gender in the HUB4 system.

3. ACROSS-WORD MODEL TRAINING

3.1. Modification of the HMM automata

When the Viterbi approximation is applied, the training of the acoustic model is based on determining the most probable path through the HMM automata of the training utterances. Part a) of Fig. 1 shows the HMM automaton for the utterance *"he is"* as used in conventional *within-word model* training. The corresponding phoneme sequences are assumed to be *"h e"* for the word *"he"* and *"i s"* for the word *"is"*. With this information (which is provided by the pronunciation lexicon) the corresponding triphones and thus the HMM automaton for the utterance can be determined. As can be seen, the context dependency at the word boundaries is reduced, all across-word contexts are denoted by a general word boundary symbol #. At the beginning of the automaton as well as after each word a one-state-HMM representing silence ($) is provided which captures optional pauses.

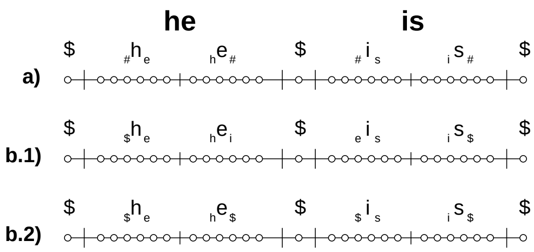

Fig. 1. HMM automaton representing the word sequence *"he is"*. a) within-word modelling; b.1) across-word modelling (word transition *with* coarticulation); b.2) across-word modelling (word transition *without* coarticulation). For simplicity the loop and the skip transition are omitted.

When *across-word models* are to be trained, the reduction of context dependency at the word boundaries is avoided. As described in [6] a word transition between two words can be modelled in two different ways:

- *word transition with coarticulation*: The acoustic models of the adjacent words influence each other.
- *word transition without coarticulation*: There is a pause between the words that is so long that the adjacent words do not influence each other.

Informal experiments have shown that the consideration of these two kinds of word transitions is crucial for good word error rate performance. Neglecting the word transition without coarticulation resulted in a significant degradation of the word error rate.
The HMM automata which realize these two kinds of word transitions are shown in Part b.1) and b.2) of Fig. 1. Part b.1) depicts the word transition *with* coarticulation. The context dependency of the phonemes is considered also across the word boundary. Part b.2) depicts the word transition *without* coarticulation. The acoustic realization of the word boundary phonemes are assumed not to influence each other, which is denoted by the special word boundary symbol $.
The type of the word transitions in the training utterances is not specified in the training transcriptions. Thus, this decision has to be made during the training procedure. In the following, two methods for making this decision are discussed.

3.2. Pause Length Estimation

This method was already mentioned in [7]. In each training iteration the lengths of the pauses between the words is extracted from the best path through the HMM automaton. The pause length between two words determines which kind of word transition is assumed in the *next* training iteration. If it is below a predefined threshold N_{sil}, the HMM automaton for a word transition *with* coarticulation (Fig. 1-b.1) is used, otherwise the automaton for the word transition *without* coarticulation (Fig. 1-b.2) is chosen. At the beginning of the training the estimation of the pause lengths has to be initialized properly [7]. In the experiments to be reported in Section 3.4 N_{sil} was chosen to be one time frame. This training method requires only minor changes to a conventional within-word model training.

3.3. Network Training

An alternative approach for across-word model training was mentioned in [4]. Both kinds of word transitions are realized in parallel in one HMM automaton as depicted in Fig. 2. In contrast to the linear sequence of HMM states used so far such a HMM automaton is called *"HMM network"* in the following.

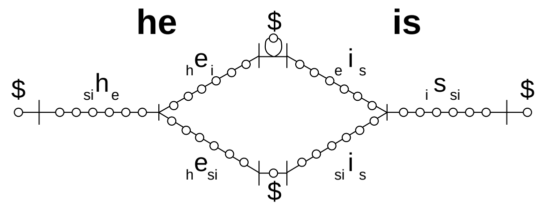

Fig. 2. Across-word HMM network representing the word sequence *"he is"*. Both kinds of word transitions are provided in parallel.

The upper path represents the word transition with coarticulation, the lower path represents the word transition without coarticulation. In both cases a silence state between the two words is provided, while the silence at the word boundary with coarticulation can be skipped without any penalty. The decision which kind of word transition has been uttered is taken during the determination of the best path through this automaton, an estimation of the lengths of the between-word pauses is not required. This method is more flexible than the pause length estimation method and can easily be modified for more enhanced pronunciation models.
However, the structure of the HMM network is more complex than the linear sequence of triphone HMMs used by the pause length estimation method. Especially the representation of one phoneme words requires four separate transitions in the HMM network as it is depicted in Fig. 3 for the training utterance *"for a while"*.

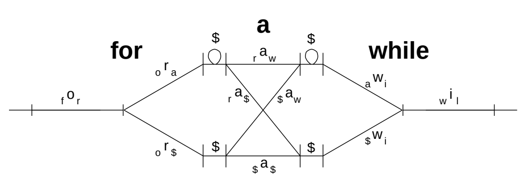

Fig. 3. Across-word HMM network for a word transition with a one phoneme word. Four separate transitions are required for the one phoneme word *"a"*. For simplicity the individual HMM states are not shown.

3.4. Results

Tab. 2 compares the active search space (average number of active states, arcs and trees per time frame after pruning) and the word error rate (WER) of both discussed training methods on NAB20k and HUB4. Since the word error rate was the main concern

in these experiments, the pruning parameters were chosen fairly conservative. The number of Gaussian densities is also given.
As can be seen, both training methods produce acoustic models which perform equally well on the two considered corpora.

Table 2. Comparison of pause length estimation training and network training on NAB20k and HUB4.

corpus	method	# dns [k]	*search space* states	arcs	trees	WER [%]
NAB20k		gender ind.				
	sil.estim.	411	15305	4251	34	11.6
	network	410	14893	4150	34	11.6
HUB4		male/female				
	sil.estim.	358 / 252	102397	28548	164	19.9
	network	358 / 251	98317	27740	160	20.1

The computational effort of the two discussed training methods was compared on NAB20k as follows: In a first experiment the training utterances were represented by linear sequences of HMM states as used in the pause length estimation method (see Fig. 1). In a second experiment HMM networks as shown in Fig. 2 and 3 were used. In both cases the computation time of one time alignment on the resulting HMM automata was measured for different model complexities, i.e. for different numbers of Gaussian densities. Exactly the same software with the same pruning setup was used. Thus, different computation times are only due to the different complexities of the HMM automata. The result is shown in Fig. 4. As can be seen, the time alignment on the HMM network requires approximately by a factor of two more time than the time alignment on a linear sequence of HMM states.

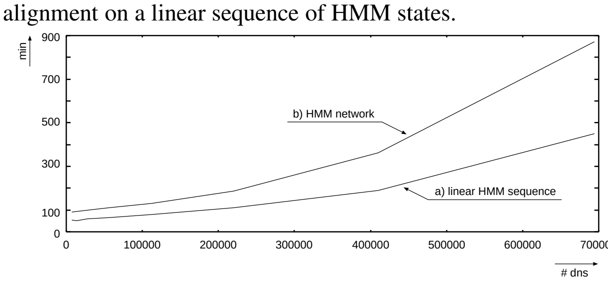

Fig. 4. Computation time for one time alignment on NAB20k as function of model complexity for a) linear HMM state sequences and b) complex HMM networks. The times were measured on an Intel Pentium III PC with 1 GHz

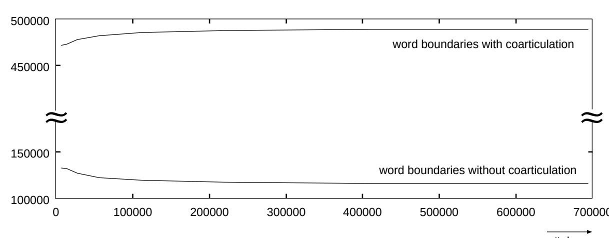

Fig. 5. Number of word transitions with and without coarticulation for different model complexities (number of Gaussian densities) on NAB20k

However, the complex HMM network used in the network training method is only needed for the decision whether a word transition has been uttered with or without coarticulation. Fig. 5 shows the number of coarticulated and not coarticulated word transitions as determined by the time alignment for different numbers of Gaussian densities on NAB20k. As can be seen, for higher numbers of densities these numbers do not change anymore. Thus, it can be assumed that for higher numbers of densities the decisions about the type of the word transitions change only rarely. Then, the complex HMM network could be replaced by a linear sequence of HMM states which represents only the needed sequence of coarticulated and not coarticulated word transitions.
This is verfied in the experiments to be reported in Tab. 3. The HMM network which allows for all combinations of coarticulated and not coarticulated word transitions was only used during the first training iterations where due to the low model complexity the time alignment on the complex HMM network is not expensive. In a first experiment the sequence of HMM states was fixed after the first split (*"Split 1"*) and the remaining iterations were performed on this linear state sequence. In a second experiment the sequence of HMM states was fixed after the third split (*"Split 3"*). The word error rate of these two systems was compared with the word error rate of the baseline system, which used the complex HMM network during the whole training (*"All Splits"*).
As the results in Tab. 3 show, keeping the state sequence fixed at the latest after the third split results in the same word error rate as the baseline system. As expected, the complex HMM network is only needed during the first iterations. Then, the sequence of the HMM states can be kept fixed and the computational effort of the remaining training iterations is similar to the pause length estimation method.

Table 3. Network training on NAB20k and HUB4. The HMM state sequence was fixed after the first and after the third split (*"Split 1"*, *"Split 3"*); the corresponding number of Gaussian densities is given in parantheses. The results are compared to the baseline system where the network was used during the whole training (*"All Splits"*).

corpus		# dns [k]	*search space* states	arcs	trees	WER [%]
NAB20k		gender ind.				
	All Splits	410	14893	4150	34	11.6
	Split 1 (14k)	409	15096	4203	35	11.8
	Split 3 (56k)	410	15231	4244	34	11.6
HUB4		male/female				
	All Splits	358 / 251	98317	27740	160	20.1
	Split 1 (8k)	355 / 248	101615	28551	163	20.1
	Split 3 (32k)	356 / 250	100328	28398	163	20.1

4. PARAMETERIZATION OF THE TRAINING

4.1. Number of tied states

As can be seen in Tab. 4, when across-word models (XW) instead of within-word models (WW) are used, the number of different triphones seen in training increases by more than a factor of two. In this section it is shown that it is advantageous to take this higher number of different triphones into account by providing a higher number of tied states.
In Tab. 5 the recognition performance of a within-word model system (WW) and an across-word model system (XW) is shown for different numbers of tied states. The optimum of this number is quite flat. For NAB20k for example it does not really matter whether 5001 or 7001 tied states are used in the XW system. However, it can be seen that the XW system needs significant more tied states for optimal recognition performance than the WW system. Compared to the baseline XW system which uses the same

number of tied states as the optimal WW system the word error rate drops down from 12.2% to 11.6% (4.9% relative) on NAB20k. On the HUB4 eval97 corpus the word error rate can be reduced from 20.7% to 19.9% (3.9% relative). Increasing the number of tied states on the WW system in the same manner does not help to improve the word error rate.

Table 4. Number of different triphones seen in training for within-word models (WW) and across-word models (XW). In case of across-word models within-word triphones are distinguished from word boundary triphones.

	WW	XW
NAB20k	9375	26239
HUB4 (male/female)	13210/13210	35635/31113

Table 5. Recognition performance for different numbers of tied states on NAB20k and HUB4. On HUB4 the same number of tied state was used for each gender.

corpus		# tied states	# dns [k]	*search space* states	arcs	trees	WER
NAB20k			gender ind.				
	WW	4001	438	5896	1777	25	12.6
		5001	508	5770	1743	25	12.8
	XW	4001	464	18265	5024	40	12.2
		5001	460	15943	4432	36	11.7
		6001	427	14956	4169	34	11.9
		7001	411	15305	4251	34	11.6
		8001	457	14781	4111	32	12.0
HUB4			male/female				
	WW	2001	330 / 230	73976	17019	105	21.2
		4001	312 / 221	77264	17394	123	21.7
	XW	2001	363 / 253	111817	30817	195	20.7
		4001	358 / 251	102397	28548	164	19.9
		6001	302 / 231	94296	26452	158	20.1
		10001	402 / 276	81877	23228	134	20.6

4.2. Position Dependency

Already in [4] it was proposed to use the *position* of a triphone within a word as an additional feature beside the phonetic context for determining the corresponding HMM when across-word models are used.
In this section the utilization of the position of a triphone as an additional piece of information during state tying is studied. The position was considered in an additional question during the construction of the decision tree [8]. It was left to the construction algorithm whether the HMM states of a word boundary triphone were tied with the states of a within-word triphone or not. In Tab. 6 three different kinds of triphone position dependency are compared on the NAB 20k corpus and the HUB4 eval97 corpus:

- *no position dependency*: the position of the triphones is not considered during construction of the decision tree.
- *simple position dependency*: across-word triphones are distinguished from within-word triphones during the construction of the decision tree.
- *complete position dependency*: as proposed in [4], four classes of triphones are considered during the construction of the decision tree: within-word triphones, triphones at the beginning of words, triphones at the end of words and triphones representing one phoneme words.

A similar approach which considered only the *simple dependency* was described in [9].

As can be seen in Tab. 6, the position dependent triphones outperform position independent triphones significantly on both corpora. *Complete* position dependent triphones perform a little bit better than *simple* position dependent triphones.

Table 6. Comparison of three different kinds of position dependency on NAB20k and HUB4.

corpus	position dependency	# dns [k]	*search space* states	arcs	trees	WER [%]
NAB20k		gender ind.				
	no	409	16501	4588	37	12.4
	simple	411	15305	4251	34	11.6
	complete	412	14926	4155	34	11.5
HUB4		male/female				
	no	358 / 251	105635	29443	170	20.4
	simple	358 / 252	102397	28548	164	19.9
	complete	357 / 252	100880	28142	159	19.7

5. CONCLUSION

In this paper we have discussed the RWTH across-word model training. Two training methods have been presented and compared with regard to recognition performance and computational effort. Furthermore, we have studied how the parameterization of the training has to be modified if instead of within-word models across-word models are to be trained.

6. REFERENCES

[1] X.L. Aubert, "One pass crossword decoding for large vocabularies based on a lexical tree search organization," in *Proc. EUROSPEECH*, Budapest, Hungary, Sep. 1999, pp. 91–96.

[2] J. J. Odell, *The Use of Context in Large Vocabulary Speech Recognition*, Ph.D. thesis, University of Cambridge, Cambridge, UK, 1995.

[3] F. Alleva, "Search organization in the whisper continuous speech recognition system," in *Proceedings IEEE Workshop on Automatic Speech Recognition and Understanding*, Santa Barbara, CA, USA, Dec. 1997, pp. 295–302.

[4] M.Y. Hwang, H.W. Hon, and K.F. Lee, "Modeling between-word coarticulation in continuous speech recognition," in *Proc. EUROSPEECH*, Paris, France, Sep. 1989, pp. 5–8.

[5] H. Ney, L. Welling, S. Ortmanns, K. Beulen, and F. Wessel, "The RWTH large vocabulary continuous speech recognition system," in *Proc. ICASSP*, Seattle, WA, USA, May 1998, vol. 2, pp. 853–856.

[6] S. Kanthak, A. Sixtus, S. Molau, and H. Ney, "Within-word vs. across-word decoding for online speech recognition," in *Proc. Automatic Speech Recognition Workshop*, Paris, France, Sep. 2000, pp. 40–45.

[7] K. Beulen, S. Ortmanns, and C. Elting, "Dynamic programming search techniques for across-word modeling in speech recognition," in *Proc. ICASSP*, Phoenix, AZ, USA, Mar 1999, pp. 609–612.

[8] K. Beulen, E. Bransch, and H. Ney, "State-tying for context dependent phoneme models," in *Proc. EUROSPEECH*, Rhodes, Greece, Sep. 1997, vol. 3, pp. 1179–1182.

[9] W. Reichl and W. Chou, "A unified approach of incorporating general features in decision tree based acoustic modeling," in *Proc. ICASSP*, Phoenix, AZ, USA, Mar 1999, pp. 573–576.