EXPERIMENTAL ANALYSIS OF THE SEARCH SPACE FOR 20 000–WORD SPEECH RECOGNITION

S. Ortmanns, H. Ney

Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology,
D-52056 Aachen, Germany

ABSTRACT

In this paper we investigate the search effort for large vocabulary continuous speech recognition. In particular, we study the effect of different pruning techniques on the search effort and on search errors. The experimental results show that it is much more efficient in the search procedure to use a tree lexicon than a linear lexicon. For the tree search method, we study the search space in detail. For the 20 000-word task under consideration, a reasonable compromise between the search effort and the recognition accuracy can be achieved by an average number of 13 000 state hypotheses per time frame. This effort is five orders of magnitude lower than the potential size of the search space. All experiments are based on our phoneme-based large vocabulary speech recognition system used in the 1994 ARPA benchmark test [?].

1. INTRODUCTION

In a series of recognition experiments, we have successfully used the so-called time synchronous beam search algorithm which is based on using word dependent copies of the pronunciation lexicon [?, ?]. This paper presents a detailed experimental analysis of the search effort and in particular a direct comparison of the linear-organized and the tree-organized search strategies. This comparison is needed for the following reason. For most tasks we use context dependent phoneme models and therefore the improvement of the tree-organized search over the linear-organized search is reduced in comparison with context independent phoneme models. In particular, when we originally introduced the tree search method [?] we use monophone models rather than triphone models. Therefore, in this paper we compare two different search strategies depending on:

- either a tree-organized pronunciation lexicon
- or a linear-organized pronunciation lexicon

For the tree-organized lexicon, we study the effect of various pruning strategies on the size of search space which is given in terms of:

- number of tree hypotheses
- number of arc hypotheses
- number of state hypotheses

This paper is divided into 3 parts. First, we begin with a review of the so-called word conditioned search algorithm. Second, we review the pruning techniques that are used in the search algorithm to avoid full search. Third, to study the search effort in detail, a series of experiments was run on the North American Business (NAB) corpus (Nov.'94).

2. THE SEARCH ALGORITHM

In this section, we review the characteristics of the search algorithm. The search procedure is based on the time synchronous beam search method as described in [?] combined with a language model look-ahead method [?]. The pronunciation lexicon can be organized in a linear fashion or in the form of a lexical tree. As the experimental results with the beam search shown [?], the lion's share of the search effort is concentrated in the initial phonemes of a word.

In the recognition experiments we use two vocabulary sizes, namely 5 000 and 20 000 words. For both vocabularies we used a set of 44 context independent phoneme models (monophones) and a set of 4688 context dependent phoneme models. The set of the context dependent phoneme models is a collection of triphones, diphones in right context and monophones [?].

By using the set of context dependent phoneme models, we have a generic tree of 63 155 phoneme arcs for the 20 000-word lexicon and 16 723 phoneme arcs for the 5 000-word lexicon, respectively. This number of arcs is to be compared with the linear-organized pronunciation lexicon. The linear lexicon consists of 123 131 phoneme copies or 29 268 phoneme copies for the 20 000-word and 5 000-word lexicon, respectively, as Table 1 shows. This leads to compression factors of 1.75 (5 000-word vocabulary) and 1.95 (20 000-word vocabulary).

In a preprocessing step, we generate a tree-organized pronunciation lexicon. Evidently, the size of the tree depends on the number of words and on the type of phoneme models. The lexicon tree contains about 500

Table 1: Number of phoneme arcs (copies) for the 5 000-word and 20 000-word vocabulary using context dependent phoneme models (CD).

Vocabulary size	Phoneme copies		Compression factor
	linear lexicon	tree lexicon	
5 000	29 268	16 723	1.75
20 000	123 131	63 155	1.95

Table 2: Distribution of the phoneme arcs over the first 6 layers of the tree lexicon for the 20 000-word vocabulary using context independent phoneme models (CI) and context dependent phoneme models (CD).

Phoneme set	Number of arcs per layer					
	1	2	3	4	5	6
CI	45	652	3799	8245	9432	7968
CD	544	3626	9335	12180	11653	9402

arcs in the first generation instead of 45 arcs when using only context independent phoneme models (monophones) as shown in Table **??**. The whole lexicon tree includes about 63 155 arcs for context dependent phoneme models and about 44 587 arcs using only monophones, respectively. For the rest of the paper we consider only the set of context dependent phoneme models.

When using a bigram language model, we face the problem that the identity of the hypothesized word w is known only when a leaf of the tree has been reached. As a result, we can apply the language model probability only at the end of a tree. To make the application of the dynamic programming principles possible, we structure the search space as follows. For each predecessor word v, we introduce a separate copy of the lexical tree as illustrated in Fig. **??**. The bold lines represent the word interior and the dashed lines represent the bigram recombination for word boundaries at time τ. For simplicity, in Fig. **??**, we omit the silence copies that are associated with each predecessor word.

To formulate the dynamic programming approach, we introduce an auxiliary quantity $Q_v(t, s; w)$ as defined in [**?**]:

> $Q_v(t, s; w)$:= probability that the best state sequence through state s of word w with predecessor word v produces the acoustic vectors $x_1 ... x_t$.

For the word interior we obtain the usual dynamic programming recursion:

$$Q_v(t, s; w) = \max_{\sigma} [\ q(x_t, s|\sigma; w)\ Q_v(t-1, \sigma; w)\ ],$$

where $q(x_t, s|\sigma; w)$ is the product of transition and emission probabilities of the acoustic models. Denoting the conditional bigram probability by $p(w|v)$ for a word pair (v, w), we have the optimization over the word boundaries:

$$Q_v(t, 0; w) = \max_{u} [\ p(v|u)\ Q_u(t, S(v); v)\ ].$$

This equation assumes that there is a special state $s = 0$ which is used to start up a word and that first the normal states $s = 1...S(w)$ are evaluated for each word w before the start-up states $s = 0$ are evaluated. $S(w)$ denotes the terminal state of word w.

3. PRUNING TECHNIQUES

Full search is prohibitive. Instead of full search, we use the time synchronous beam search strategy, where at each time frame only the most promising hypotheses are retained. The pruning approach consists of three steps that are performed each 10-ms time frame [**?**]:

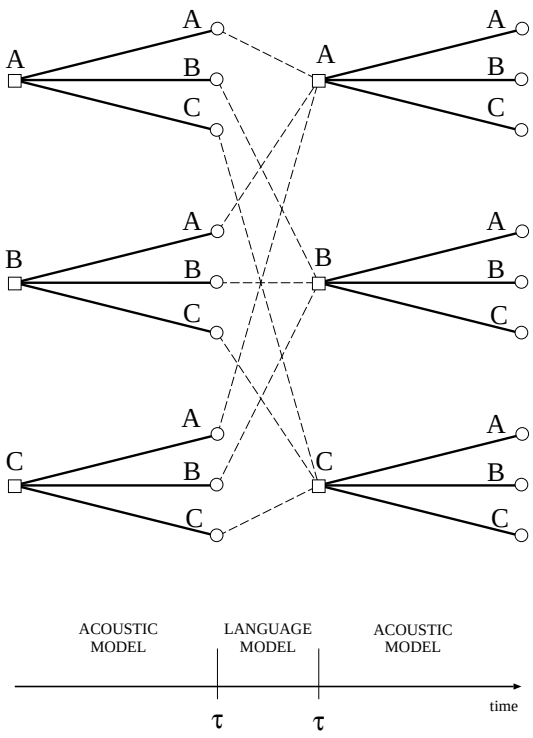

Figure 1: Illustration of the bigram language model recombination at time τ for lexicon trees.

- *Histogram pruning.* This method limits the maximum number of surviving state hypotheses. The pruning parameter will be referred to as the maximum number of active states (*MaxHyp*).
- *Acoustic pruning.* Only hypotheses with a score relatively close to the best state hypothesis are retained for further considerations. This so-called beam width, i.e. the number of surviving state hypotheses, is indirectly controlled by the so-called acoustic pruning threshold (*AcuThr*).
- *Language model pruning.* This method is applied only to hypotheses of tree start-ups as follows. At word ends, the bigram probability is incorporated into the accumulated score, and the best score for each predecessor word is used to start-up the corresponding tree. The scores of these tree start-up hypotheses are now subjected to an additional pruning step which, in principle, is similar to the acoustic pruning. However, there is a separate pruning parameter which is referred to as language model pruning threshold (*LanThr*).

4. EXPERIMENTAL RESULTS

4.1. Recognition Task

The experiments were carried out on the Wall Street Journal (WSJ) and on the North American Business (NAB) corpora. The training of the acoustic models was performed on WSJ 0 and WSJ 1 training data as described in [**?**]. In the experiment we used about 290 000

mixture densities for each gender. For the analysis of the search space, the experimental test conditions can be summarized as follows:

- We used a subset of the NAB'94 H1 development data which contained 155 sentences with 3671 spoken words from 10 male speakers.
- 2.9% (108 spoken words) of the spoken words were out-of-vocabulary words, i. e. they were not part of the 20 000-word vocabulary.
- The language model was a bigram model with a test set perplexity of $PP_{bi} = 216$.

4.2. Comparison: Linear vs. Tree Lexicon

In an initial experiment, we tested the beam search method for both the linear lexicon and the tree lexicon. For this comparison we used only a subset of the WSJ Nov.'92 development data and the 5 000-word vocabulary, since the computational cost for the linear-organized search is extremly high. This subset contained 130 spoken words of the 6 male speaker. The test set perplexity PP_{bi} of the bigram language model was 196. Table ?? gives an overview of the search space in terms of the average number of state hypotheses per time frame, the average number of word ends per time frame and the word error rate.

Although the compression factor between the tree and linear lexicon is less than 2, the tree lexicon leads to a reduction of the number of state hypotheses by a factor of more than 14. This reduction factor is much higher than the compression factor of the lexicon, since most of the search effort is spent on the first 3 phonemes of a word. As a result, for a 20 000-word vocabulary, a linear organization of the lexicon is highly inefficient. Hence, for the following we consider only the tree search method.

4.3. Histogram Pruning

Next, we investigated the recognition accuracy as a function of the search effort which depends on the pruning parameters. First, we report the experimental results on the histogram pruning. In these experiments we kept the other two pruning parameters fixed. In an informal experiment these two parameters ($AcuThr = 110\,000, LanThr = 70\,000$) had been adjusted beforehand. Table ?? shows that, by increasing the parameter *MaxHyp* from 50 000 to 200 000, the reduction in the number of search errors was negligible.

Table 3: Comparison: linear vs. tree lexicon on a subset of the WSJ Nov.'92 development data (6 male speakers, 130 spoken words, $PP_{bi} = 196$).

Lexicon	States	Word ends	DEL-INS	WER[%]
linear	81 090	1185	5-5	17.6
	125 144	2036	5-1	10.7
tree	9 050	159	4-1	8.5

Table 4: The effect of limiting the maximum number of state hypotheses on the word error rate (with the parameters $AcuThr = 110k$ and $LanThr = 70k$).

max. number of States (*MaxHyp*)	average number of			WER[%]
	States	Arcs	Trees	
25 k	13008	3639	43	18.4
50 k	17862	4943	51	18.3
75 k	20383	5611	54	18.3
100 k	21894	6008	56	18.3
125 k	22852	6257	57	18.3
150 k	23511	6428	57	18.2
200 k	24338	6640	58	18.2

4.4. Acoustic and Language Model Pruning

Using a maximum number of 100 000 state hypotheses, we studied the effect of the acoustic pruning threshold (*AcuThr*) on the search effort and on the recognition accuracy. Table ?? depicts the maximum and average number of active states, of active arcs, of active trees and of word ends per time frame after pruning. In addition, the last column gives the word error rates.

As shown in Table ??, we varied the acoustic pruning threshold between 50 000 and 170 000 and observed that good results can be achieved with a value of 100 000. In these experiments the number of state hypotheses varied between 252 and 79 836. Looking at the word error rate, we can see that a reasonable compromise is achieved by 13 000 state hypotheses, which result in a word error rate of 18.4 %. To further reduce the word error rate to 18.1 %, we have to increase the average number of state hypotheses by a factor of 6.2.

Choosing the acoustic pruning threshold $AcuThr = 100\,000$ in Table ??, we can make the following observations:

- There are only 38 tree copies instead of 20 000 for full search.
- On the average, there are 238 ending words (rather than $20\,000^2$).
- As a result, the 238 word end hypotheses are recombined into 38 tree start-ups. In other words, on the average the search process produces 6 to 7 word end hypotheses for the same acoustic word, but with different predecessor words.

In another series of experiments, we analyzed the effect of the language model pruning threshold. The results are shown in Table ?? and indicate that for values $LanThr = 50\,000$ to 110 000 there is no significant effect on the word error rate. However, we can observe a slight reduction of the search space.

The experiments show that, in order to keep the number of search errors at a reasonable level, the pruning parameters have to be adjusted for a 20 000-word vocabulary as follows:

- The maximum number of active states (*MaxHyp*)

Table 5: Search space (active states, arcs, trees, word ends) and word error rates in [%] after pruning for different acoustic pruning thresholds (with the parameters $LanThr = 65k$, $MaxHyp = 100k$).

AcuThr	States		Arcs		Trees		Word ends		DEL–INS	WER[%]
	ave.	max.	ave.	max.	ave.	max.	ave.	max.		
50 k	252	4873	80	1237	4	56	9	289	181 - 297	45.6
60 k	677	16312	213	4167	6	116	17	632	127 - 176	28.3
65 k	1068	31568	332	7994	9	150	24	942	115 - 140	24.2
75 k	2396	99851	703	25825	15	277	49	2145	105 - 121	20.6
100 k	12908	99998	3554	35849	38	525	238	6328	96 - 105	18.4
110 k	21720	99996	5943	36895	49	613	388	6352	95 - 105	18.3
120 k	32538	100000	8838	36333	59	611	564	6389	95 - 105	18.2
130 k	43862	100000	11784	36332	67	617	745	6361	95 - 105	18.2
170 k	79836	100000	20649	36137	87	613	1363	7078	95 - 105	18.1

should be set to a value of 100 000.

- The acoustic pruning threshold (*AcuThr*) should be chosen so that the average number of state hypotheses at each time frame is about of 13 000.
- The language model pruning threshold (*LanThr*) should be chosen so that the average number of trees is about 40.

For such a choice of pruning parameters, the size of the actual search space is 13 000 state hypotheses as compared to the full size of the potential search space which is 20 000 trees $\cdot$ 65 000 arcs $\cdot$ 6 states $= 7.56 \cdot 10^9$ states. Using this set of pruning parameters, beam search required 70 times real time on a SGI workstation (Indy R4600). The response time was equally divided into search and likelihood calculations (for 290 000 mixture densities).

5. SUMMARY

This paper reported experimental tests with beam search on the WSJ/NAB task. The results are summarized as follows:

- We compared two different search strategies depending on a tree-organized lexicon and a linear-organized lexicon. We found that the word conditioned tree search reduced the size space by more than a factor of 14.
- For the tree search algorithm, we studied the interdependence of the recognition accuracy and the search effort as a function of the pruning parameters. As the experiments show, for the 20 000-word task, good recognition results can be achieved for an average number of state hypotheses of about 13 000 states and 40 trees per time frame, respectively. This effort is five orders of magnitude lower than the potential size of the search space.

Table 6: The effect of the language model pruning parameter (*LanThr*) on the search effort and on the word error rate (with the parameters $AcuThr = 110k$, $MaxHyp = 100k$).

LanThr	average number of				WER[%]
	States	Arcs	Trees	Word ends	
25 k	16008	4148	11	271	18.6
50 k	20522	5532	31	352	18.3
65 k	21720	5943	49	388	18.3
110 k	22119	6115	101	446	18.3

REFERENCES

1. X. Aubert, C. Dugast, H. Ney, V. Steinbiss: "Large Vocabulary Continuous Speech Recognition of Wall Street Journal Corpus", In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. II, pp. 129-132, Adelaide, Australia, April 1994.
2. C. Dugast, R. Kneser, X. Aubert, S. Ortmanns, K. Beulen, H. Ney: "Continuous Speech Recognition Tests and Results for the NAB'94 Corpus", In *Proc. ARPA Spoken Language Technology Workshop*, Austin, TX, pp. 156-161, January 1995.
3. H. Ney, D. Mergel, A. Noll, A. Paeseler: "Data Driven Organization of the Dynamic Programming Beam Search for Continuous Speech Recognition", IEEE Trans. on Signal Processing, Vol. SP-40, No. 2, pp. 272-281, Feb. 1992.
4. H. Ney, R. Haeb-Umbach, B.-H. Tran, M.Oerder: "Improvements in Beam Search for 10000-Word Continuous Speech Recognition", In *Proc. Int. Conf. on Acoustics, Speech and Signal Processing*, San Francisco, CA, Vol. I, pp. 9-12, March 1992.
5. H. Ney, V. Steinbiss, R. Haeb-Umbach, B.-H. Tran, U. Essen: "An Overview of the Philips Research System for Large-Vocabulary Continuous-Speech Recognition", International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 1, pp. 33-70, 1994.
6. H. Ney, V. Steinbiss, X. Aubert, R. Haeb-Umbach: "Progress in Large Vocabulary Continuous Speech Recognition", In *Proc. in Artificial Intelligence, Progress and Prospects of Speech Research and Technology*, Munich, pp. 75-92, September 1994.
7. V. Steinbiss, B.-H. Tran, H. Ney: "Improvements in Beam Search", In *Proc. Int. Conf. on Spoken Language Procesing*, Yokohama, pp. 2134-2146, September 1994.