EFFICIENT EVALUATION OF THE LVCSR SEARCH SPACE USING THE NOWAY DECODER

Steve Renals

Department of Computer Science
University of Sheffield
Sheffield S1 4DP, UK
S.Renals@dcs.shef.ac.uk

Mike Hochberg

Nuance Communications
333 Ravenswood Ave.
Menlo Park CA 94025, USA
mmh@coronacorp.com

ABSTRACT

This work further develops and analyses the large vocabulary continuous speech recognition (LVCSR) search strategy reported at ICASSP-95 [1]. In particular, the posterior-based phone deactivation pruning approach has been extended to include phone-dependent thresholds and an improved estimate of the least upper bound on the utterance log-probability has been developed. Analysis of the pruning procedures and of the search's interaction with the language model has also been performed. Experiments were carried out using the ARPA North American Business News task with a 20,000 word vocabulary and a trigram language model. As a result of these improvements and analyses, the computational cost of the recognition process performed by the NOWAY decoder has been substantially reduced.

1. INTRODUCTION

At ICASSP-95, we introduced an efficient search procedure [1] that was implemented as a software decoder known as NOWAY and used in the ABBOT hybrid connectionist/ HMM LVCSR system [2, 3]. Key features of this approach included both likelihood- and posterior-based pruning, a stack ordered by hypothesis reference time and log-probability, and a tree-structured lexicon. In particular, it was found that the posterior-based pruning approach, *phone deactivation pruning*, was extremely effective and offered up to an order of magnitude speed improvement with little or no search error.

In this paper, we present further analysis of the search algorithm and improved approaches to both likelihood- and posterior-based pruning. Results are presented from the ABBOT system using a 20,000 word vocabulary, trigram language model and recurrent network acoustic model trained using the SI-84 data. Most of the experiments reported used the ARPA 1993 spoke 5 development test set (Sennheiser microphone).

This work was performed while Mike Hochberg was at Cambridge University Engineering Department and was partially supported by ESPRIT project 6487, WERNICKE.

2. THE NOWAY START-SYNCHRONOUS DECODER

The search strategy adopted in NOWAY may be described as *start-synchronous* stack decoding[1]. Either the Viterbi criterion or the full likelihood (forward probability) criterion may be used. All the experiments reported here used the Viterbi criterion.

Stack decoding is a heuristic search technique which avoids an exhaustive evaluation of the search space. The search algorithm usually operates by removing the highest scoring hypothesis from a priority queue of partial and complete hypotheses and extending it by one word. This requires the comparison of hypothesis scores at different times, which may be achieved by using a heuristic estimate of the minimum cost of extending a partial hypothesis to a valid complete hypothesis. If the heuristic gives a guaranteed overestimate of the score (underestimate of the cost), then the search is admissible[2].

This best-first approach can be problematic as it requires looking ahead. In common with Bahl and Jelinek [4], Paul [5] and Gopalakrishnan *et al.* [6], the partial hypotheses are ordered by length and the shortest is extended first. This may be conveniently represented by a set of priority queues: one for each time. Rather than looking ahead, an estimate of the least upper bound $\mathrm{lub}P(t_h)$ on the log-probability P_h of a hypothesis h with reference time t_h is employed as a reference for likelihood-based pruning.

For computational efficiency, the pronunciation lexicon is represented as a tree. For the ABBOT system, cross-word modelling is not required and a single tree suffices. Both likelihood- and posterior-based approaches to pruning are used. The likelihood-based approaches are based on beam search and involve the definition of a pruning beamwidth and the setting of a maximum number of partial hypotheses that may be extended at any time. (Note that limiting the maximum number of partial hypotheses that may be extended can be regarded as an adaptive reduction of the beamwidth.) The posterior-based approaches are more specific to systems with a posterior probability estimator (*e.g.*, hybrid connectionist/HMM systems such as ABBOT). These approaches are described below.

[1] The term "start-synchronous" was suggested to us by John Bridle (personal communication).

[2] This approach is also known as A^* search.

20K trigram						
	Phones Pruned (%)		Spoke 5		Eval 1992	
			Decode	Word	Decode	Word
Approach	Correct	Total	Time	Error	Time	Error
PI (7.5e-5)	0.1	69	2.4	12.1	2.3	13.0
PD	0.1	64	1.9	11.8	1.8	13.3
PI (5e-4)	0.4	84	1.0	12.4	1.0	13.6
PD	0.4	79	1.0	13.0	1.0	13.9
PI (3e-3)	1.1	91	0.4	15.8	0.4	16.1
PD	1.1	87	0.6	16.7	0.6	15.6

Table 1: Comparison of phone-dependent (PD) and phone-independent (PI) phone deactivation pruning thresholds. The terms in brackets indicate the threshold employed for the PI case. The phone-dependent thresholds were developed on the 1993 spoke 5 devtest set, and recognition experiments were carried out both on this data and the 1992 evaluation set. Decode time is in multiples of realtime on a pentium-based PC.

3. PHONE DEACTIVATION PRUNING

Posterior-based phone deactivation pruning has proved extremely successful in increasing the search efficiency of hybrid connectionist/HMM systems [1, 7]. This section presents an empirical analysis of the behaviour of this technique and extends it to incorporate phone-dependent thresholds.

For each frame of acoustic data, the connectionist model produces a complete vector of context-independent posterior phone probabilities. These phone posteriors may be regarded as a local estimate of the presence of a phone at a particular time (frame). If the posterior probability estimate of a phone given a frame of acoustic data is below a threshold, then all words containing that phone at that time frame may be pruned (deactivated), *i.e.*,

$$\begin{array}{ll} \textbf{if} & P(\text{phone}_i|\text{data}) \ < \ \text{threshold}_i \\ \textbf{then} & P(\text{phone}_i|\text{data}) \ := \ 0. \end{array}$$

This process is referred to as *phone deactivation pruning*. In the zero probability case, converting to log likelihoods would involve evaluating log(0), so in practice the (scaled) log likelihood is set to a large negative value.

Results for when the threshold was constant across all phones (*i.e.*, $\text{threshold}_i = \text{threshold}, \forall i$) were reported in [1]. The posterior probability threshold used to make the pruning decision may be empirically determined using an operating curve derived from a development set. In figure 1, such a curve is illustrated by plotting the fraction of "correct" phones pruned[3] versus the fraction of total phones pruned as the threshold is varied over its complete range. This operating curve demonstrates that phone deactivation pruning substantially reduces the search space. At the threshold value of 7.5e-5, 0.1% of the correct phones have been pruned (virtually no search errors are observed) while 70% of the phones have been removed from the search.

[3]The fraction of correct phones pruned is automatically determined using a Viterbi alignment procedure.

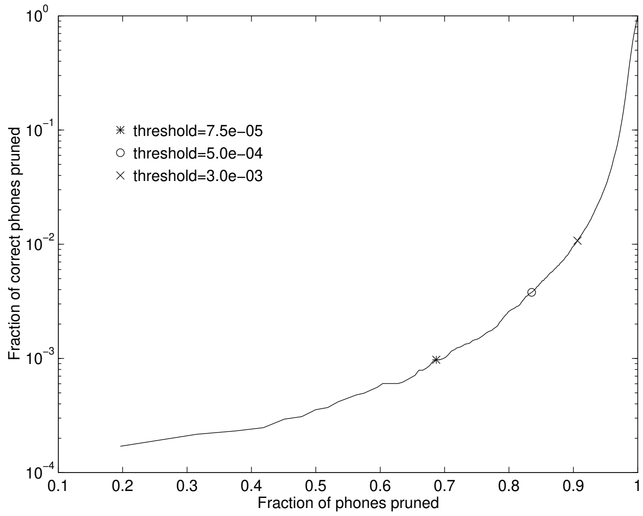

Figure 1: Operating curve for determination of the phone deactivation pruning threshold (phone-independent).

We have recently investigated the use of phone-dependent thresholds in phone deactivation pruning. This work was motivated primarily by the fact that choosing an operating point does not reflect the pruning for any particular phone. For example, the threshold of 5e-4 in figure 1 gives an average correct phones pruned rate of 0.4% while the actual range across phones varies from 0% to 50%. Our initial approach to determining the phone-dependent thresholds has been to specify a constant phone deactivation error rate, ϵ, on a development test, *i.e.*,

$$\text{threshold}_i = \max x \in (0,1) : cp_i(x) < \epsilon$$

where $cp_i(x)$ is the correct phone i pruned rate given the threshold x. Table 1 shows a comparison between the two approaches. These preliminary results are inconclusive, perhaps due to the small development set used to estimate the phone-dependent thresholds. Further experiments are in progress.

4. LANGUAGE MODEL INTERACTION

The NOWAY search procedure usually incorporates exact LM scores only at the word ends. If LM probabilities are not incorporated within words[4] the within-word log-probabilities will be larger than the end of word log-probabilities which incorporate LM information. Using this information, the pruning beamwidth within a word can be set more tightly than at a word end, resulting in increased efficiency with minimal search errors incurred (table 2). As well as using the decode time as a measure of the effective size of the search space, three other criteria were also used: the average number of nodes activated per start time (*i.e.*, per tree traversal); the average number of hypotheses created per

[4]An upper bound on the LM probabilities is used for all words in a given context. This bound can be precomputed in advance.

1993 spoke 5 devtest, 20K trigram						
Parameters		Performance		Search Space		
WE beam	WW beam	Decode Time	Word Error	Nds Actd.	Hyps Crtd.	Hyps Extd.
9	9	7.7	12.1	714	84	8.2
9	8	5.5	12.1	517	72	8.2
9	7	3.6	12.3	350	60	8.1
8	8	5.0	12.2	516	42	6.0
8	7	3.3	12.4	349	35	6.0
8	6	2.1	12.2	219	30	5.9

Table 2: Effect of varying the within-word beamwidth (WW-beam) given a constant word-end beamwidth (WE-beam) of 8.0 or 9.0. Decode time is in multiples of realtime on a pentium-based PC.

start time; and the average number of hypotheses extended per start time (*i.e.*, the average stack size after pruning).

Experiments in which exact LM probabilities were incorporated within words have been carried out and the results are summarized in table 3. In these experiments, the control parameter was a threshold relating to the number of pronunciations passing through a node in the tree. When the number of pronunciations represented by a node was less than a threshold, the exact LM score (or the maximum of a set of LM scores) was incorporated into the within-word probabilities. In addition, *hypothesis pruning* was employed to prune individual elements from the set of hypotheses being extended in parallel. As shown in the table, the narrower beam for within-word pruning causes search errors when LM information is incorporated within words. To fairly evaluate the effect of incorporating LM information into the within-word probabilities, no within-word incorporation of LM information is compared with reduced within-word beam case (WE-beam = 4, WW-beam = 2) and varying degrees of LM incorporation with WE-beam = WW-beam = 4. The method introduced in section 5 was used to compute lub$P(t_h)$ in these experiments.

These results indicate that the extra computation (caused by the extra number of LM accesses) is not counter-balanced by a significant reduction in the search space. The results in the table also indicate that individual pruning of hypotheses does not result in a more efficient search—and indeed can result in an increase in search errors.

Table 4 gives further details on the statistics of LM access and indicates that LM incorporation within words leads to an increased amount of back-off computation.

5. LEAST UPPER BOUND ESTIMATION

The approximated A^* heuristic used in the search algorithm relies on an accurate estimate of lub$P(t_h)$. This bound needs to be initialized and updated as new hypotheses are extended. An accurate estimate of lub$P(t_h)$ allows a narrower beam width to be employed without increasing search errors. Note that when using a long-span (*e.g.*, trigram) language model, the probability backtrace of the most probable complete hypothesis will not necessarily coincide with lub$P(t_h)$ when $t_h < T$.

1993 spoke 5 devtest, 20K trigram							
Parameters		Performance			Search Space		
Thresh/ HP?	Beam WE/WW	Time	Error	Search Error	Nds Actd.	Hyps Crtd.	Hyps Extd.
0/F	4/2	2.4	12.1	0.0	172	66	3.6
0/F	4/4	3.5	12.2	0.8	460	107	3.7
0/T	4/2	2.2	12.8	5.5	253	40	3.3
0/T	4/4	6.0	12.2	0.8	707	69	3.5
1/F	4/2	2.2	12.9	6.3	253	49	3.4
1/F	4/4	5.8	12.3	1.7	707	88	3.6
1/T	4/2	2.3	12.8	5.5	253	30	3.3
1/T	4/4	6.0	12.2	0.8	707	69	3.5
10/F	4/2	1.8	13.3	9.4	232	28	2.9
10/F	4/4	4.5	12.3	1.7	600	60	3.6
10/T	4/2	1.7	14.1	15.6	232	17	2.6
10/T	4/4	4.5	12.7	5.0	602	35	3.2
2500/F	4/2	6.2	17.5	42.2	69	22	3.3
2500/F	4/4	7.1	12.7	5.0	147	38	3.5
2500/T	4/2	5.0	19.4	56.2	72	11	2.7
2500/T	4/4	5.7	15.5	28.1	146	20	3.0

Table 3: Effect of within-word incorporation of exact language model probabilities. If the number of words whose pronunciations pass through a node is equal to or less than "Thresh" the exact LM probability is used. If the "HP?" predicate is true, individual hypotheses are pruned at individual nodes as they are extended in parallel through the tree. All experiments were carried out on a pentium-based PC and time is in multiples of realtime. The maximum number of pronunciations passing through any node was 2061.

Previously, a fairly crude estimate for lub$P(t_h)$ was used. A simple "garbage model" was used to generate the initial estimates. This approach averaged the likelihoods of the N most probable phones (ranked by posterior probability and excluding the most probable) and combined the result with a nominal Markov process score. The estimate of lub$P(t_h)$ was then updated whenever a frame of a proposed partial extension to a hypothesis exceeded the current least upper bound.

Although this method of updating lub$P(t_h)$ requires no additional computation, it is suboptimal. In particular, the constraints of the pronunciation dictionary and language model are not used to provide a better estimate. To incorporate this information into the search, a technique similar to that used in the envelope search of Gopalakrishnan *et al.* [6] was developed. The key idea is that lub$P(t_h)$ is only updated using paths obtained from backtraces of complete word extensions. This involves more computational effort than the previous method and requires more memory since backtrace information must be stored. However, a much tighter estimate of the least upper bound is obtained using this approach. The resultant estimate of lub$P(t_h)$ for a typical sentence is shown in figure 2 with the log-probability trace of the most likely complete hypothesis. The initialization procedure based on the garbage model was retained.

We have experimented with this method of least upper bound estimation and compared it with the previous approaches. Results (using the same measures of search

1993 spoke 5 devtest, 20K trigram			
Average LM accesses per frame			
	No LM/2	No LM/4	LM/4
Trigrams	168	180	180
Bigram backoffs	31	75	117
Unigram backoffs	93.6	256	456
Cache accesses	1912	6361	7631
Decode Time	2.3	6.3	5.8
Word Error %	12.1	12.2	12.3

Table 4: Effect of LM incorporation in the search. Results are for incorporation at word ends only using a WW-beam of 2 or 4 (No LM/2 and No LM/4) and incorporation within-words at nodes that are part of one pronunciation only (LM/4). WE-beam was 4 for all experiments. Decoding times are multiples of realtime on a pentium-based PC (note that these are slower than in table 3 due to the overhead of collecting LM access statistics).

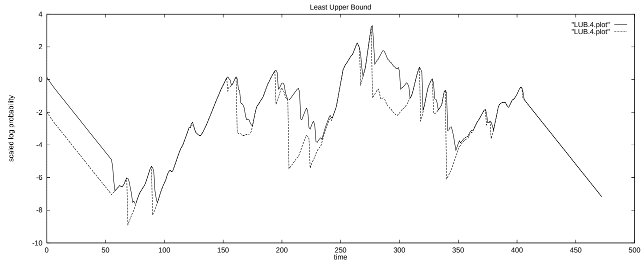

Figure 2: Least upper bound estimate. The upper, solid line is the estimate of lub$P(t_h)$ using the backtrace method; the dashed line is the probability of the most probable complete hypothesis.

space size as above) are summarized in table 5. Additional experiments in which phone deactivation pruning was not applied gave even larger performance improvements—using the spoke 5 data the decoding time in this case was reduced from 50× realtime using the old lub estimation to less than 10× realtime using the new lub estimation. These results indicate that this method of lub estimation produces a significant reduction in the size of the search space without adversely affecting the word error rate. This general trend has been observed in further experiments using different datasets [8].

6. CONCLUSION

We have analysed and developed the search strategy introduced in [1]. The behaviour of phone deactivation pruning was empirically investigated and a version using phone-dependent thresholds was introduced. An improved method of estimating lub$P(t)$ resulted in a substantial reduction in the volume of the search space evaluated to achieve similar word error rates. The effect of the language model on the search was analysed and the results used to derive more efficient pruning thresholds.

1993 spoke 5 devtest, 20K trigram						
Parameters		Performance		Search Space		
WE Beam	WW Beam	Decode Time	Word Error	Nds Actd.	Hyps Crtd.	Hyps Extd.
lub: Greedy estimate						
9	9	7.7	12.1	714	84	8.2
9	**8**	**5.5**	**12.1**	**517**	**72**	**8.2**
9	**7**	**3.6**	**12.3**	**350**	**60**	**8.1**
8	8	5.0	12.2	516	42	6.0
8	7	3.3	12.4	349	35	6.0
8	6	2.1	12.2	219	30	5.9
lub: Backtrace estimate						
5	5	9.9	12.1	655	184	5.5
5	4	6.8	12.1	457	156	5.5
5	3	4.6	12.1	296	129	5.5
4	4	3.5	12.2	460	107	3.7
4	**3**	**4.1**	**12.1**	**297**	**87**	**3.7**
4	**2**	**2.4**	**12.1**	**172**	**66**	**3.6**
3	3	3.7	12.4	297	50	2.0
3	2	2.0	12.4	171	38	2.0
3	1.5	1.6	12.8	124	31	2.0
3	1	1.0	13.3	84	24	1.9

Table 5: Comparison of the new (backtrace) approach to least upper bound estimation with the previous (greedy) approach. Lines in bold indicate the optimal settings of the beamwidths for efficient recognition with minimal search error based on these and other experiments [8].

7. REFERENCES

[1] S. Renals and M. Hochberg, "Efficient search using posterior phone probability estimates," in *Proc. ICASSP '95*, pp. 596–599, 1995.

[2] A. J. Robinson, M. M. Hochberg, and S. J. Renals "The use of recurrent networks in continuous speech recognition," in *Automatic Speech and Speaker Recognition*, C. H. Lee, F. Soong and K. K. Paliwal (eds.), Kluwer, 1996.

[3] M. M. Hochberg, S. J. Renals, A. J. Robinson, and G. D. Cook, "Recent improvements to the ABBOT large vocabulary CSR system," in *Proc. ICASSP '95*, pp. 69–72, 1995.

[4] L. R. Bahl and F. Jelinek, "Apparatus and method for determining a likely word sequence from labels generated by an acoustic processor." US Patent 4,748,670, May 1988.

[5] D. B. Paul, "An efficient A* stack decoder algorithm for continuous speech recognition with a stochastic language model," in *Proc. ICASSP '92*, pp. 25–28, 1992.

[6] P. S. Gopalakrishnan, L. R. Bahl, and R. L. Mercer, "A tree-search strategy for large vocabulary continuous speech recognition," in *Proc. ICASSP '95*, pp. 572–575, 1995.

[7] S. Renals, "Phone deactivation pruning in large vocabulary continuous speech recognition," *IEEE Signal Processing Letters*, to appear.

[8] S. Renals and M. Hochberg, "Decoder technology for connectionist large vocabulary speech recognition". Technical Report CS-95-17, Department of Computer Science, University of Sheffield, 1995. (`ftp://ftp.dcs.shef.ac.uk/share/spandh/pubs/renals/cs-95-17.ps.gz`)