Effect of Task Complexity on Search Strategies for the Motorola Lexicus continuous speech recognition system

Sreeram V. Balakrishnan
Motorola Lexicus Division, 3145 Porter Drive, Palo Alto, CA 94304, sree@lexicus.mot.com

1.0 Abstract

As speech recognition systems are increasingly applied to real world problems, it is often desirable to use the same recognition engine for a variety of tasks of differing complexity. For example the recognizer in a dictation system may need to handle a highly constrained correction grammar, as well as a large vocabulary dictation trigram. This paper explores the relationship between the complexity of the recognition task and the best strategies for pruning the recognition search space.

We examine two tasks: 20000 word WSJ dictation at the complex end, and phone book access using a 60 word grammar at the 'simple' end. All experiments are conducted using a tied state, continuous density, cross-word, triphone HMM system, with a time-synchronous, single pass recognizer [1] [2] . For both tasks we compare two strategies for pruning the search space: absolute pruning, and rank based pruning. In absolute pruning, the number of hypotheses is controlled by eliminating ones that have a score less than a fixed beamwidth below the best scoring hypothesis [2] [3] . In rank based pruning, the hypotheses are ranked by score and all hypotheses beneath a certain rank are eliminated [2] .

We first present a description of the system used to perform the recognition task, along with details of the tasks themselves. Next, a set of statistics characterizing the behavior of the recognizer under different pruning regimes, will be presented for each task. By analyzing these, we will show how the different strategies will have different speed versus error-rate trade offs. Finally, we present results comparing the error-rates resulting from the different pruning strategies.

2.0 System Description

All recognition experiments were performed using a single-pass Viterbi recognizer with cross-word triphone HMMs. The search is implemented by maintaining a set of hypotheses $Q = \{q_1 \ldots q_n\}$, where each hypothesis contains a word history $h = h(q)$, a pronunciation network node id $n = n(q)$ and a score $s = s(q)$. Associated with each word history is a context $c = c(h)$, which consists of the last $n-1$ words, where, n is the order of word n-gram language model (if one is being used). In the case of large vocabulary recognition, the pronunciation network is actually a pronunciation tree, and associated with each terminal leaf is a list of words. For constrained vocabulary tasks where the grammar is a set of BNF rules, the pronunciation network is obtained by first compiling the BNF rules into a network with arcs corresponding to words, and then replacing the words with their pronunciation.

Assuming an utterance is divided into T frames of speech $f_1 \ldots f_T$, the search implemented by the recognizer proceeds as follows[1] [3] [4] . The recognizer starts at time $t = 1$, with a set of hypotheses Q that has a single start hypothesis q. It then loops through the speech frames, updating the scores of the hypotheses in Q with each new frame. Hypotheses that pass a propagation pruning beamwidth **propagate**(q) are allowed to propagate to the child nodes of the hypothesis. The new hypotheses are put in the set Q_{new}; in addition, hypotheses that are potential word ends are inserted in the set Q_{word}. The hypotheses in Q_{word} are then checked to see if they pass a word propagation pruning beamwidth **propagate_word**(q). If they do, then they are allowed to start new word trees, and the resulting hypotheses are also inserted into Q_{new}. All the hypotheses in Q are tested to see if they pass a survival pruning beamwidth, **survive**(q). If they do, they are inserted into Q_{new}. Next, all hypotheses in Q_{new} that have the same context $c(h(q))$ and node $n(q)$ are merged, by removing the hypothesis with the lower score. Finally, Q_{new} becomes Q for the next loop.

The Pseudo Search Code in Table 1 summarizes the above algorithm. The pruning of the search is controlled by the

true/false functions: **propagate**(q), **propagate_word**(q), and **survive**(q). We explore two forms of these functions. In the first case, absolute pruning, a beamwidth is set in terms of the difference in score between a hypothesis and the best scoring hypothesis in the set. If this difference is below a preset beamwidth, then the function returns true. In the second case, rank pruning, the hypotheses are ranked according to their score. Higher scores will have lower ranks, and if a hypothesis has a rank lower than a preset rank, the function returns true.

TABLE 1. Pseudo Search Code for Recognizer

for $t = 1 \ldots T$

Update_Scores (Q, f_t)

Propagate_Children (Q, Q_{word}, Q_{new})

Propagate_Words (Q_{word}, Q_{new})

Prune (Q, Q_{new})

Merge (Q_{new})

$Q = Q_{new}$

end

3.0 Task description

The two tasks chosen to investigate the different pruning strategies are at opposite ends of the complexity spectrum. The first task, is large vocabulary dictation using a 20k vocabulary and trigram backoff language model. Both the vocabulary and trigram are based on the November 1992 ARPA WSJ evaluation. A left cross word triphone system was built using the WSJ0 training set (SI 84, 7200 training utterances), with 30K Gaussians. The test set consisted of 333 sentences drawn from the November 1992 evaluation test set.

The second task, is constrained dialog phone book access. The vocabulary consists of 60 words and the set of possible utterances is constrained by a set of BNF rules. These rules allows phrases such as:

please exit to main menu

main menu please

business listings please

please cancel

The BNF rules are compiled into a network with 380 nodes, with each node having on average between 3 and 4 word arcs. This word network is then converted into a pronunciation network by replacing the words with their pronunciations.

The system was trained on the ATIS sentences of Macrophone, and left cross word triphone models with 30K gaussians were built. We collected 74 test utterances from 3 different speakers recorded over a telephone line. The complexity of this task is very similar to command and control, and other menu based dialog systems, and as such was taken as proxy for the behaviour of such systems.

4.0 Analyzing Search Efficiency

An ideal pruning strategy will prune out all hypotheses except the one that leads to the word sequence with the highest probability. Referring to this hypothesis as the aligned hypothesis, we can characterize how efficient absolute and rank pruning strategies are, by recording the score and rank of the aligned hypothesis relative to the best scoring hypothesis at each time step during recognition. We define search efficiency in terms of the average number of hypotheses per frame of speech, or $mean(|Q|)$. Assuming that the amount of computation per hypothesis is constant, $mean(|Q|)$ represents the total amount of computation required to update, propagate and prune the hypothesis set.

To obtain the data on the score and rank of the aligned hypotheses, we first ran the recognizer using absolute pruning and adjusted the beamwidths to be slightly wider than needed to obtain the best error-rate. An initial run was performed for each task, to generate detailed frame to HMM state alignments for the final word sequences obtained by the recognizer. A second run was then performed, during which the score and rank of the aligned hypothesis were tracked for each frame of speech.

FIGURE 1. Difference between best and aligned hypothesis score vs. % frames with lower difference

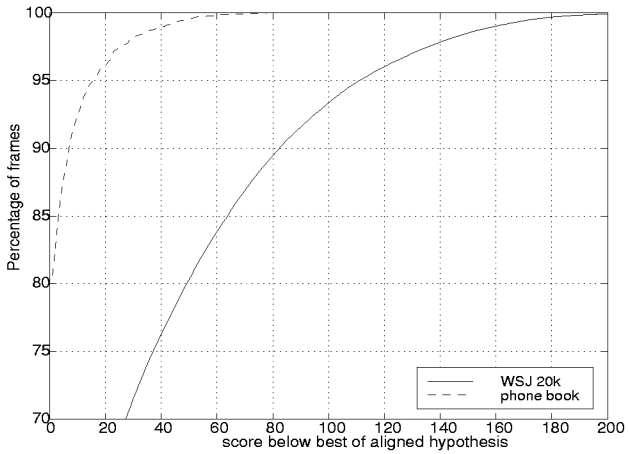

Figure 1 plots the cumulative percentage of frames of test speech versus the difference in score between the best scoring hypothesis and the aligned hypothesis, for both tasks. Figure 2 shows the cumulative frame percentage versus the rank of the aligned hypothesis, for just the phone book task, and Figure 3 is the same as Figure 2 except for the WSJ 20k task.

FIGURE 2. Rank of aligned hypothesis versus % of test frames with lower rank for phone book access

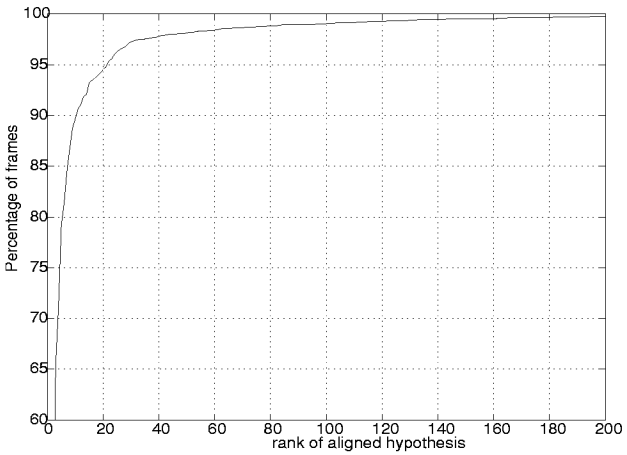

FIGURE 3. Rank of aligned hypothesis versus % of test frames with lower rank for WSJ 20k dictation

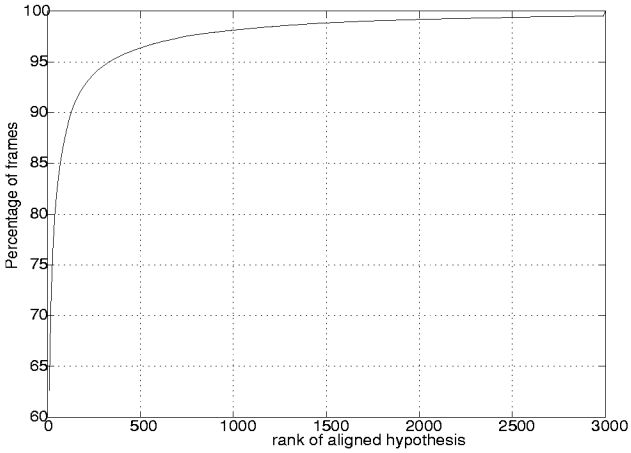

Table 2 shows the actual error-rate obtained for each task along with the value of $mean(|Q|)$ and the absolute beamwidth used for **survive**(q), with a) showing the results for beamwidths set to the level used in generating Figures 1-3. The results in b) and c) show that any reduction of the beamwidths beyond the level in b) causes an increase in error-rate. This agrees well with the plots shown in Figure 1, since for both tasks, the beamwidth of **survive**(q) in b) corresponds to the point where 100% of the test frames would survive being pruned, whereas for c) a small fraction of the aligned frames start to get pruned out.

The plots of the rank of the aligned hypothesis reveal a different story. For the case of phone book access, 99.5% of the test frames have an aligned rank less than 150, and for WSJ 20k the same 99.5% mark is reached at rank 3000. According to Table 2 b), with optimal absolute pruning, $mean(|Q|)$ for the phone book task is 146, which is close to 150, whereas the $mean(|Q|)$ value for WSJ 20k is 10900. This actually corresponds to the point at which only 57 frames out of 122,246 or 0.05% have a higher aligned rank. Thus there may be considerable scope to tighten the search in the case of WSJ 20k by using rank based pruning.

Further justification for this supposition can be drawn from Figure 4. In addition to tracking the rank and score of the aligned path, we also track the number of the hypotheses with a score above the best hypothesis minus the **survive**(q) beamwidth. We term this the beamwidth rank, and the ratio of the rank of the aligned path to the beamwidth rank is a measure of search inefficiency, since all hypotheses with a rank greater than the aligned rank and less than the beamwidth rank, are wasted computation. Figure 4 shows plots of this ratio for both tasks.

TABLE 2. Summary of Results using absolute pruning

		Phone Book	WSJ 20K
a)	word error rate	10.7%	13.3%
	$mean(\|Q\|)$	186	12600
	beamwith of **survive**(q)	90	220
b)	word error rate	10.7%	13.3%
	$mean(\|Q\|)$	146	10900
	beamwith of **survive**(q)	80	200
c)	word error rate	11.8%	13.8%
	$mean(\|Q\|)$	129	9400
	beamwith of **survive**(q)	70	180

Clear differences for each task are apparent. For the WSJ 20k task, 98% of the test frames have a ratio less than 0.1, while the equivalent percentage for the phone book task is only reached at a ratio of 0.4. Thus absolute pruning seems

to be far more inefficient for WSJ 20k than it is for phone book access.

To verify this, a final set of experiments were conducted, in which absolute pruning was combined with rank based pruning. **survive**(q) was modified so that it would return true only if the score of q was within an absolute beamwidth of the best and the rank of q was less than a fixed rank beamwidth. The absolute beamwidths were initially set to be the same as Table 2 b), and the rank beamwidths were set as tight as possible without affecting the error-rate. The absolute beamwidths were then increased to the point that they played no part in pruning, leaving only the rank beamwidth as the effective pruning mechanism. The results are summarized in Table 3, with a) showing the results using absolute pruning with effective beamwidths, and b) the results with the absolute beamwidths widened to the point it was ineffective.

FIGURE 4. Ratio of aligned path rank to rank of hypothesis with score = best - survive(q) beamwidth, versus % frames with lower ratio.

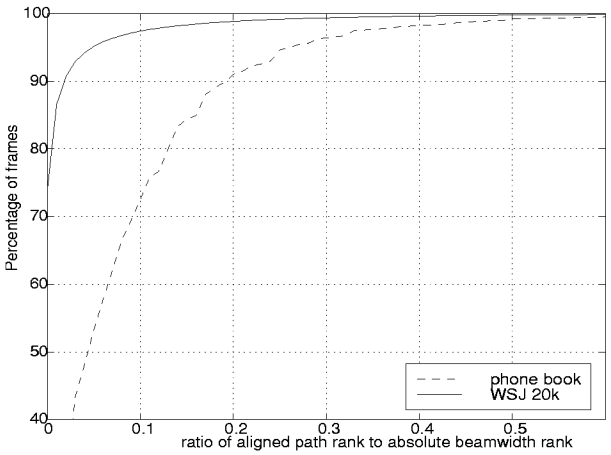

TABLE 3. Results of combining rank based pruning with absolute pruning for survive (q)

		Phone Book	WSJ 20K
a)	word error rate	10.7%	13.3%
	$mean(\|Q\|)$	100	5450
	rank beamwidth	200	4000
	absolute beamwidth	80	200
b)	word error rate	10.7%	13.3%
	$mean(\|Q\|)$	184	6400
	rank beamwidth	200	4000
	absolute beamwidth	800	800

For both tasks, the addition of rank based pruning allows a substantial increase in efficiency without compromising accuracy. However, the gains are much greater for the WSJ 20k task, with a reduction in $mean(|Q|)$ of 50%, versus a reduction of only 30% for the phone book task. In addition, Table 3 b) shows that when only rank pruning is being used, the error-rate does not change, but $mean(|Q|)$ increases by 84% for the phone book task, whereas it only increases by 17% for the WSJ 20k task. This indicates that the absolute pruning beamwidth plays a larger role in pruning the search space for the phone book task. Comparing the results in Table 2 b) with Table 3 b), it appears that if a choice has to be made between rank or absolute pruning, then absolute would be more efficient for the simple phone book access task, while rank based pruning would be more effective for the large vocabulary task.

5.0 Conclusions

For both constrained dialog phone book access, and WSJ 20k large vocabulary recognition, combining rank based pruning with absolute pruning provides a significant gain in search efficiency over only absolute pruning. However, the gains are much more significant for the more complex large vocabulary task. If only one pruning strategy can be implemented, then absolute pruning is more efficient for constrained dialog tasks such as the phone book access task. In contrast, rank based pruning is much more efficient for the highly complex WSJ 20k dictation task.

1. J.Odell, V. Valtchev, P.C.Woodland, S.J.Young: A One-Pass Decoder Design for Large Vocabulary Recognition. ARPA Spoken Language Technology Workshop, Plainsboro, NJ, pp. 405-410, March 1994
2. J. Odell: Use of Context in Large Vocabulary Speech Recognition, Ph.D Thesis, March 1995, University of Cambridge, Engineering Department.
3. H. Ney, S. Ortmanns: Progress in Dynamic Programming Search for LVCSR. Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, pp287-294.
4. Fil Alleva: Search Organization in the Whisper Continuous Speech Recognition System. Proceedings of 1997 IEEE Workshop on Automatic Speech Recognition and Understanding, pp295-302.