REDUCING PEAK SEARCH EFFORT USING TWO-TIER PRUNING

Mark Wright, Simon Hovell and Simon Ringland

markw@saltfarm.bt.co.uk

Speech Technology Unit, BT Laboratories,
Martlesham Heath,
Suffolk, England.

ABSTRACT

Many of the pruning strategies used to remove less likely hypotheses from the search beam in large vocabulary speech recognition (LVR) systems, have a peak search space many times greater than the average search space.
This paper discusses two such pruning strategies used within BT's speech recognition architecture [1], *Step pruning* and *Histogram pruning*.
Two-tier pruning is proposed as a simple but powerful extension applicable to either of the above strategies. This seeks to limit the expansion of the search space between the prune and acoustic match processes without affecting accuracy. It is shown that the application of two-tier pruning to either strategy reduces peak search effort, and results in an average reduction in run time of 33% and 53% for step pruning and histogram pruning respectively, with no loss in top-N accuracy.

1. INTRODUCTION

Large vocabulary speech recognition (LVR) systems incorporate many thousands of words and have high perplexities. This results in very large decoding networks. In order to make decoding of such networks practical, only the more likely hypotheses are considered by constraining the search to a beam within the network. The width of this beam as recognition progresses greatly affects both the accuracy and search time of the decoder. One of the established ways of constraining search space is to use a pruning algorithm [2]. At time t, the pruning algorithm compares the log probabilities, or scores, $S_{hy}(t)$, of all N partial path hypotheses, $h_y(t)$ (where $1 \leq y \leq N$), with a threshold, $T(t)$, and prunes those where

$$S_{h_y}(t) \leq T(t) \qquad (1)$$

This leaves a beamwidth, $S_{hB}(t) - T(t)$, where $S_{hB}(t)$ is the score of the best partial path hypothesis. The question naturally arises of how best to derive $T(t)$. Conventionally, pruning has used a simple fixed threshold based pruner, or a step pruner where the threshold is varied dynamically with time [3,4]. A typical such approach is discussed in section 2.

A common approach to large vocabulary tasks is to use dynamic network recognisers, where the grammar network is created on the fly, rather than being predefined and loaded in advance [5, 1]. In such a design, pruning is crucial to prevent exponential network growth. Also, maintaining a consistent search space can significantly speed up recognition for a given accuracy. Histogram pruning, discussed in section 3, is a more sophisticated technique than step pruning, and provides a level of control that moves these types of task closer to real-time performance [6]. These algorithms work by reducing the *existing* search space. In section 4, an extension to these methods is introduced, two-tier pruning, which seeks to control the *expansion* of the search space during the course of the decoding process. This significantly reduces the gap between the peak and the average search space, although there is still scope for further improvement. The final section of this paper presents results on a set of experiments carried out using these techniques on two different tasks and shows the advantages and disadvantages of each technique.

2. STEP PRUNING

A convenient way to describe the operation of a typical step pruning algorithm is in terms of a token passing paradigm [7]. In this paradigm, tokens representing a particular partial path hypothesis traverse the recognition network during decoding. At the end of recognition, the winning hypothesis is extracted by tracing back the path taken by the best scoring token.
Step pruning is used to maintain control over the search space via a dynamic pruning beam, and is described as follows:

- Take the score of the current best scoring partial path hypothesis and subtract from this some factor, $F(t)$.
- $F(t)$ is set by inspecting the search space, $N(t)$.
- If the current search space is currently bigger than that desired, then reduce $F(t)$ by a fixed step size, K, making the beamwidth smaller. Hypotheses must therefore have scores closer to the best hypothesis to avoid pruning.
- If, however, $N(t)$ is smaller than desired, $F(t)$ is increased by the step size and the increased beamwidth allows correspondingly worse scoring hypotheses to survive.
- $F(t)$ is restricted to vary between two predefined limits, that is, $F_{max} > F(t) > F_{min}$.

Thus, the threshold is derived by:

$$F(t+1) \equiv \begin{cases} Max(F(t)-K, & F_{min}) & N(t)>N_D \\ Min(F(t)+K, & F_{max}) & N(t)<N_D \\ F(t) & & N(t)=N_D \end{cases} \quad (2)$$

where N_D is the desired search space, and

$$T(t+1) = S_{h_B}(t+1) - F(t+1) \quad (3)$$

Using the above paradigm allows some control of search space and a typical use of resources is shown in figure 1.

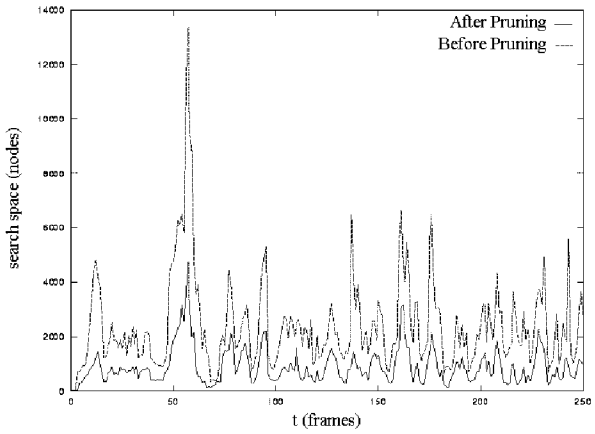


Figure 1: Search space using step pruning during recognition of the utterance *"The board meets the last week of this month"*, showing the large difference in pre- and post-prune search effort.

Whilst this dynamic approach to threshold setting is a distinct improvement to the use of a static threshold, it still suffers from disadvantages. As can be seen in figure 1 and the above algorithm description, little account is taken of the amount by which the search space prior to pruning is too big or small. Control is limited since the threshold can only change by a fixed step size during each stage of the input. Often this results in too small a beamwidth for a large step size and therefore pruning of well matching hypotheses. Alternatively, with a smaller step size, the beamwidth frequently fails to keep pace with the rapidly expanding search space leading to a marked drop in speed performance. For large unconstrained tasks, as mentioned earlier, this is particularly true. The somewhat erratic control available using step pruning can be greatly improved using the histogram paradigm described in the next section.

3. HISTOGRAM PRUNING

A more precise method of constraining the search space to the desired size is to employ a score frequency distribution graph. At each time t of the input, a score frequency histogram (such as that shown in figure 2) is compiled. This can be used to calculate accurately the pruning threshold needed in order to maintain the search space at a constant level.

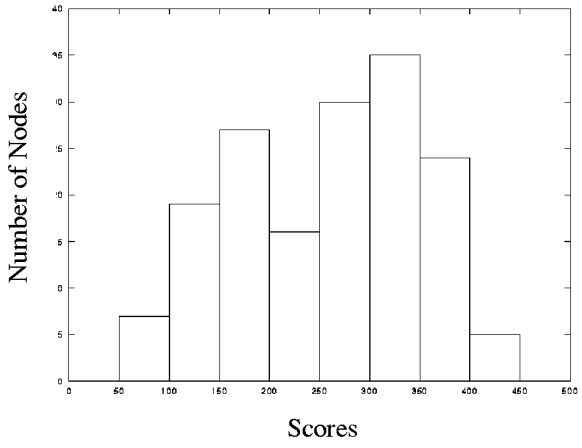


Figure 2: Score frequency Histogram.

Thus if a histogram $P(t)$, comprising M bins, is populated with all $N(t)$ hypotheses scores, where the population of bin i is defined as $p_i(t)$ and

$$\sum_{i=0}^{M-1} p_i(t) = N(t) \quad (4)$$

with the range of scores populating the ith bin ranging from

$$Sp_i(t) \text{ to } Sp_{i+1}(t) \quad (5)$$

the threshold $T(t)$ is calculated as

$$T(t) = Sp_m(t) \quad (6)$$

where m = the minimum value of m such that

$$\sum_{i=0}^{m} p_i(t) \geq N_D \quad (7)$$

and N_D is the desired search space.

The improvement in control, over the step pruning paradigm, is shown in figure 3. Here it is seen that the search space at each time t is much closer to that desired (in this case 1000 nodes) than for the step pruning case. The observed magnitude of deviations from the desired search space, is dictated by the quantisation due to the width (given in equation 5) of each bin.

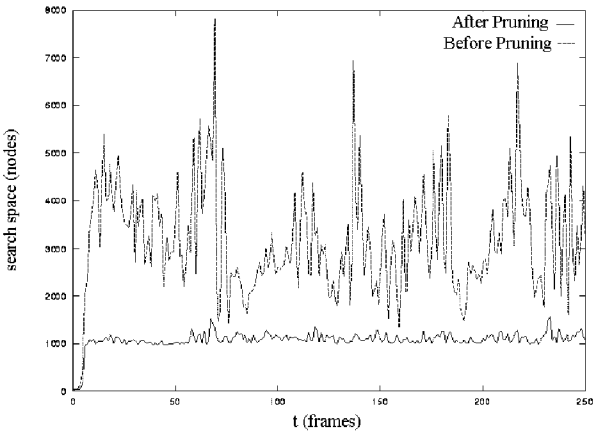


Figure 3: Histogram pruning with a desired search space of 1000 nodes. Search effort during recognition of the utterance *"The board meets the last week of this month"*.

A problem existing with both the step and histogram paradigms, is that the search space that was constrained by pruning expands again prior to the next acoustic match and subsequent pruning cycle. This was highlighted by Steinbiss et al [6] and is observed in both figures 1 and 3. In figure 3 for example, the search space remaining after pruning expands again by up to 8 times as partial path hypotheses are extended. This expanded network contains those nodes upon which the next acoustic match (a computationally expensive process) must be performed, and defines the *pre-prune* search space. Two-tier pruning helps to contain expansion of the network between pruning cycles, and therefore restricts the *pre-prune* search space size, making it closer to that remaining directly after pruning (*post-prune* size).

4. TWO-TIER PRUNING

The network expansion problems highlighted above, occur due to hypotheses extending into unlikely parts of the network. As hypotheses extend further along their partial paths, the new nodes that the paths extend through are included in the search effort. In practice, this extension usually happens between the prune and acoustic match processes. This results in the search space constraint imposed by the pruner being increased to an undesired level. This problem is redressed to some extent by the use of two-tier pruning. Under this paradigm, all nodes with current hypotheses are considered as comprising two parts (similar to that described by Young et al [7]):

The first, the *model hypotheses*, are the best scores from the hypotheses of each acoustic model represented by the nodes. These scores can be represented by $S_{hy}(t)$ and use equation 6 in order to define a first tier beam by which model hypotheses are pruned.

The second part, hypotheses capable of propagation to following nodes are termed *exit hypotheses* and are controlled using a second, smaller, beam. This second beam thus applies to those hypotheses which are about to extend a partial path or about to bring previously pruned nodes in an existing path back into the search space. In this way new nodes are only added to the search space if they are reasonably likely to survive. This approach avoids bringing nodes into the search space only to remove them again immediately during subsequent pruning. The resultant improvement in pre- and post-prune search space difference can be seen by comparing figures 3 and 4 where the input utterances are identical. This improvement consequently brings peak search effort closer to average effort for the overall recognition process.

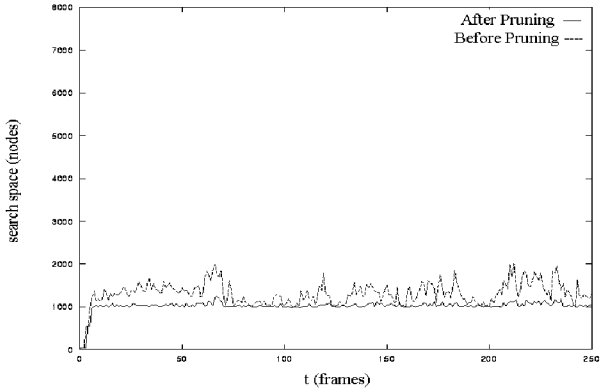


Figure 4: Two-tier histogram pruning. Search effort during recognition of the utterance *"The board meets the last week of this month"* showing a reduction in pre-prune search space.

For two-tier histogram pruning, a separate histogram is maintained and populated with the exit hypotheses mentioned above. The threshold defining the second beamwidth is found by following a similar mechanism to that detailed in equations 4 to 7. When applying a two-tier paradigm to step pruning, maintenance of an independent set of the parameters detailed in section 2 is required (apart from the term $S_{hB}(t)$ which is common to both tiers). Again, exit hypotheses are pruned via a smaller beam derived from these parameters and a similar mechanism to that described in section 2 is used. The effect on the pre- and post-prune search space difference is shown in figure 5.

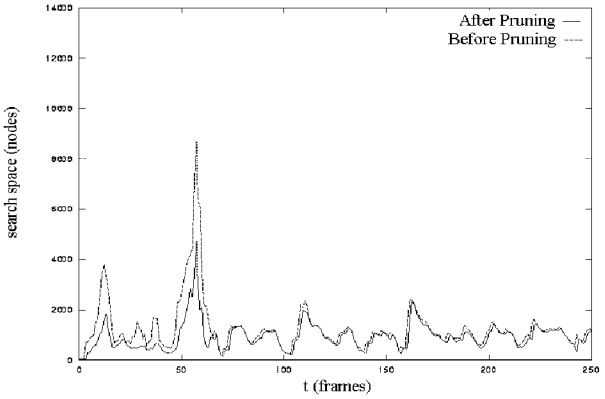
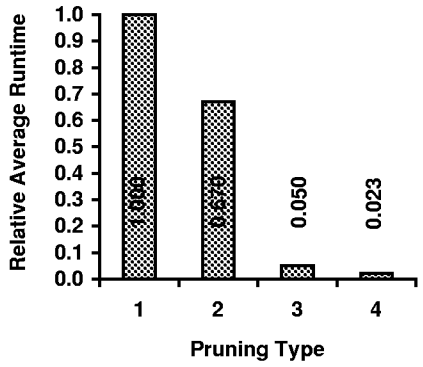


Figure 5: Two-tier step pruning. Search effort during recognition of the utterance *"The board meets the last week of this month"* showing a reduction in pre-prune search space over figure 1.

Again, it can be seen by comparison of figures 5 and 1, that use of a two-tier paradigm reduces the search space expansion after pruning. Peak and average search space for the utterance are therefore brought closer together. Furthermore, the envelope of the post-prune search space remains largely unchanged by application of the second tier.
The extra control gained from two-tier pruning leads to significant speed improvements for LVR systems. A series of experiments detailing the relative performance of each pruning paradigm is presented in the next section.

5. RESULTS

Two-tier pruning is aimed at combating search space problems in LVR systems. The investigations presented here use a system having a 5000 word vocabulary. The experiments carried out involve recognition of a subset of utterances from the WSJCam0 test set [8] . The average recognition times for each of the paradigms presented above can be seen in figure 6. Speed performance is given as a proportion of the worst performing paradigm which was single-tier step pruning.



- 1. Single-tier step pruning
- 2. Two-tier step pruning
- 3. Single-tier histogram pruning
- 4. Two-tier histogram pruning

Figure 6: Speed performance as a function of pruning mode for a similar accuracy.

The results show a significant improvement in speed performance between step and two-tier step pruning. The transition between single and two-tier pruning for the histogram case also shows a large reduction in runtime. The slowest paradigm for the experiment shown was single-tier step pruning. A reduction in runtime of 97.67% was achieved by moving from single-tier step to two-tier histogram pruning.
The results presented in figure 6 are for a large vocabulary system. Note that for small vocabulary systems, step pruning will still outperform histogram pruning as the relative computational complexity of the latter outweighs the benefits of the extra control that it brings.

6. CONCLUSION

The problem of controlling the search space in LVR systems has been examined. Step pruning was found to provide a limited degree of control over the size of the search space provided the severity of the pruning was set at a suitable level. It was then shown that histogram pruning affords much more precise control of the post-prune search space than step pruning. However, on its own, even histogram pruning fails to control the expansion of the search space between pruning cycles. It has been shown that the introduction of a second tier of pruning can largely contain this expansion, bringing the peak search effort for a recognition much closer to that of the average.
From the results presented in section 5, it can be seen that the control offered by two-tier histogram pruning leads to a large reduction in recognition time over single-tier step pruning. This is the case on LVR systems, whereas for small vocabularies step pruning is more speed efficient. This is due to the extra computational requirements in determining the pruning threshold for the histogram case.
It has been shown that two-tier pruning, applied to either the step or histogram paradigms, offers significantly more control over the search space without loss in accuracy. This gives a reduction in processing time of 33% for step pruning and 53% for histogram pruning.

7. REFERENCES

[1] Hovell S. A., Ringland S.P.A and Ollason D.G.: "A Modular Architecture for Speech Recognition", Proc 1995 IEEE ASR Workshop, Snowbird, pp 189-190, 1995.

[2] Ney H: "Architecture and Search Strategies for Large-Vocabulary Continuous-Speech Recognition", Proc NATO-ASI, Bubion, Spain, pp 59-84, 1993.

[3] Haeb-Umbach R, Ney H: "Improvements in Time-Synchronous Beam Search for 10000-Word Continuous Speech Recognition", IEEE Trans Speech and Audio Processing, Vol 2, pp353-356, 1994.

[4] Steinbiss V: "Sentence-Hypotheses Generation in a Continuous Speech Recognition System" Proc European Conference of Speech Communication and Technology, Vol 2, pp 51-54, Paris, 1989.

[5] Woodland PC., Odell JJ, Valtchev V, Young SJ: "Large Vocabulary Continous Speech Recognition Using HTK", Proc. ICASSP Australia, pp 125-128, 1994.

[6] Steinbiss V, Tran BH, Ney H. "Improvements in Beam Search". Proc ICSLP Japan, VOL 4 pp 2143 -2146, 1994.

[7] Young SJ, Russell NH, Thornton JHS: "Token Passing: A Simple Conceptual Model for Connected Speech Recognition Systems". Technical Report CUED/F-INFENG/TR38, Cambridge University Engineering Dept, 1989.

[8] Fransen J,Pye D, Robinson AJ, Woodland PC, Young SJ: "WSJCAM0 corpus and recording description", Technical Report CUED/F-INFENG/TR192, Cambridge University Engineering Dept, 1994.