New pruning criteria for efficient decoding

Janne Pylkkönen

Neural Networks Research Centre
Helsinki University of Technology, Finland
janne.pylkkonen@hut.fi

Abstract

In large vocabulary continuous speech recognizers the search space needs to be constrained efficiently to make the recognition task feasible. Beam pruning and restricting the number of active paths are the most widely applied techniques for this. In this paper, we present three additional pruning criteria, which can be used to further limit the search space. These new criteria take into account the state of the search space, which enables tighter pruning. In the speech recognition experiments, the new pruning criteria were shown to reduce the search space up to 50% without affecting the search accuracy. We also present a method for optimizing the threshold parameters of the pruning criteria for the selected level of recognition accuracy. With this method even a large number of different pruning thresholds can be determined with little effort.

1. Introduction

Decoding efficiency continues to be a very important issue in automatic speech recognition (ASR) systems. The ever increasing computing power is utilized by more and more complex models of acoustics and language in order to cut down the gap between the accuracy of ASR systems and that of a human. Therefore, the efficiency needs to be considered as a problem of its own.

In practice, the decoding can be viewed as a search of an optimal path in a large search network. The search network comprises the constraints set by the different knowledge sources, and it can be constructed either statically before the recognition or dynamically during the recognition. In almost any case, an exhaustive search through the network is intractable in large vocabulary continuous speech recognition (LVCSR) tasks [1]. To make the search feasible, we need to compromise the optimality of the search by introducing pruning.

Ideal pruning restricts the search space effectively without degrading the recognition accuracy. In this respect, the basic beam pruning works quite well. It alone can provide enough restrictions to the search space to make the recognition feasible, and the pruning threshold can be chosen to minimize the number of search errors due to pruning [2]. The desired level of pruning depends on the recognition task, and the pruning thresholds are usually determined by optimizing the recognition accuracy on a separate development data set.

In this paper, we propose three new pruning criteria which can be used in addition with the basic prunings to further limit the search space. We also present a method for easily determining the pruning thresholds for these and other criteria. The pruning methods are evaluated in two LVCSR tasks.

This paper concentrates only on the pruning of the search space, not e.g. to the pruning of the Gaussians in acoustic probability computations. The methods presented are intended for one-pass time-synchronous decoders (see e.g. [1]), although they can be applied to other decoding techniques too.

2. Basic pruning methods

The idea of the search space pruning is to retain only the most promising path hypotheses as the starting points for the following path expansions. The relative goodness of the paths can be determined by their likelihood scores. Following the notations in [3], we denote as $Q_w(t,s)$ the overall likelihood score of the best partial path that ends at time t in state s of the search network with word history w.

2.1. Global beam pruning

The so called beam search is probably the most important pruning criteria used in LVCSR decoders. In this paper, we refer to it as global beam pruning, as it can be applied in all states of the search network. The global beam pruning retains only paths with a likelihood score close to the best partial path hypothesis [3]. More formally, we can define the likelihood score of the best partial path as

$$Q_{GB}(t) = \max_{(v,\sigma)}\{Q_v(t,\sigma)\}. \qquad (1)$$

The pruning criterion then states that those paths are pruned for which

$$Q_w(t,s) < f_{GB} \cdot Q_{GB}(t). \qquad (2)$$

The threshold f_{GB} determines the width of the beam. If likelihood scores are stored in logarithmic domain, the beam pruning criterion becomes

$$\log Q_w(t,s) < \log Q_{GB}(t) - f'_{GB}, \qquad (3)$$

where $f'_{GB} = -\log f_{GB}$ is a positive beam width. Figure 1 shows a conceptual example of this pruning criterion.

The beam pruning method needs to know the likelihood score of the best partial path hypothesis at each time frame. It is still possible to apply a first level of pruning already when expanding the path hypotheses, using the best partial path which has already been expanded as an estimate for the best path. To achieve maximal pruning it is then necessary to add a further pruning step after the actual best likelihood score is available, but the early pruning is effective enough to benefit the efficiency despite a small computational overhead.

2.2. Histogram pruning

Another common pruning criterion is the histogram pruning, which limits the number of active paths at each time frame by retaining only a predefined number of best paths [3]. The name histogram pruning is used because the pruning can be done efficiently using a histogram of likelihood scores. Compared to

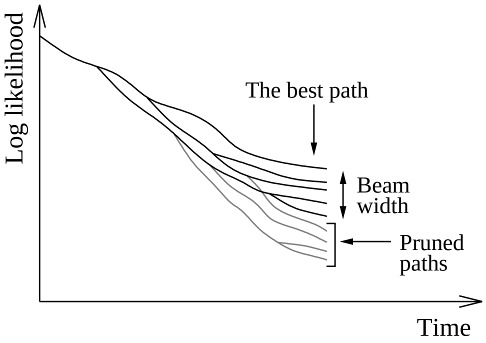

Figure 1: *A conceptual example of the global beam pruning.*

the global beam pruning histogram pruning has the benefit of defining a worst case processing time for decoding. In many cases, however, it is not as effective as the global beam pruning, so these two criteria are often used together.

2.3. Word end pruning

The two pruning criteria described above are applied to all path hypotheses. Besides these global prunings, it is also possible to define prunings specific to the state of the search space which a search hypothesis occupies. One pruning criterion that is often reported is applied after the language model score has been added to the likelihood score of the path hypothesis. In [3] this pruning is called the language model pruning, but we refer to it as the word end pruning, as it is used at the end of a lexical search network, where the word identity has been resolved. The idea is the same as with the global beam pruning, except that we compare the likelihood scores to the best partial path hypothesis in the word end position and use a tighter beam width.

Let us define the likelihood score of the best word end hypothesis as

$$Q_{WE}(t) = \max_{(v,\sigma \in S_{WE})} \{Q_v(t,\sigma)\}, \tag{4}$$

where S_{WE} is the set of word end states of the search network. The pruning is then applied to those path hypotheses, which are in one of the word end states and a path is removed if its likelihood score fulfills

$$\log Q_w(t,s) < \log Q_{WE}(t) - f'_{WE}, \tag{5}$$

where f'_{WE} is the corresponding beam width.

The word end pruning is useful for two reasons. The global beam width must be wide enough to tolerate the addition of the language model score. The word end beam width can therefore be tighter, because for the paths this pruning is applied to the language model score has been added recently. Another advantage of the word end pruning is that usually the search network has a high level of branching in these word end positions, due to beginning of a new word. The paths at those positions are therefore likely to be expanded to numerous new path hypotheses, so it is beneficial to prune them more tightly.

2.4. Setting the pruning thresholds

The pruning thresholds can be used to adjust the tradeoff between the recognition accuracy and efficiency. If accuracy is to be maximized, the tightest possible pruning thresholds that still give the optimal accuracy can be found by evaluating different values iteratively using a development data set.

Besides fixed pruning thresholds, it is also possible to adaptively change them during decoding. This is useful especially for the global beam width when histogram pruning is also used. As stated above, the global beam pruning (and also the word end pruning) can be applied already when the best likelihood score of the current time frame is not yet available. Histogram pruning, on the other hand, can be used only after all path hypotheses have been expanded. It would be beneficial if the paths which will be pruned due to histogram pruning could be pruned already before expansion. This situation can be approximated by adjusting the global beam width based on the number of expanded path hypotheses in previous frames. One method for this was presented in [4].

3. New pruning criteria

Extending the idea behind the word end pruning, it is possible to define additional specific pruning criteria, which use separate reference likelihood scores and pruning thresholds. Next we present three new pruning criteria which we show to give performance boosts in the evaluation.

3.1. Equal depth pruning

If the search network is organized as a lexical prefix tree (see [1] for more information about the tree organization of the search network), it is easy to define for each state a depth from the root of the tree as the number of states between the state and the root. Path hypotheses ending at the states at equal depth may have common properties which enable the use of a tighter beam threshold. We therefore define the *equal depth pruning* as follows:

$$Q_{ED}(t,s) = \max_{(v,D(\sigma)=D(s))} \{Q_v(t,\sigma)\}, \tag{6}$$

$$\log Q_w(t,s) < \log Q_{ED}(t,s) - f'_{ED}, \tag{7}$$

where $D(s)$ denotes the depth of the state s, and f'_{ED} is the beam width of the pruning.

In our decoder, the depths of the search network states are most naturally computed at the level of HMM states. The number of depth levels is therefore slightly too fine grained, so we divide the depth value by two and retain only the integral part.

3.2. Equal word count pruning

Adding language model scores to the path likelihood score causes discontinuities, which at worst may throw an otherwise feasible path hypothesis outside the global beam. The situation is problematic especially if the differences between the word counts of the competing path hypotheses are large, implying considerable differences in the added language model likelihood scores. The global beam width may therefore need to be rather wide, so some efficiency is lost in situations where the word counts do not differ.

To improve the prunings in these situations, we define the *equal word count pruning*, for which the reference likelihood score is relative to the word count of the word history. The pruning criterion can be stated as

$$Q_{EWC}(w,t) = \max_{(C(v)=C(w),\sigma)} \{Q_v(t,\sigma)\}, \tag{8}$$

$$\log Q_w(t,s) < \log Q_{EWC}(w,t) - f'_{EWC}, \tag{9}$$

where $C(w)$ denotes the number of words in the word history, and f'_{EWC} is the beam width.

3.3. Fan-in pruning

One more specific pruning criterion was defined for path hypotheses which are at the so called fan-in states. These states model the context dependent phones at the beginning of the words. In our decoder, these states are shared among all the words, so it is important to be able to prune the paths in these positions as effectively as possible before they get expanded to different word beginnings.

4. Optimizing the pruning thresholds

The pruning thresholds are usually optimized using a development data set to evaluate the performance with different threshold values. The search of the optimal thresholds can be done by simply trying different values iteratively, or using, for example, some form of a line search scheme. But as the number of different pruning criteria increases, it soon becomes difficult to quickly obtain optimal values for all the thresholds.

To optimize our pruning thresholds described in the previous sections, we use the following procedure. The global beam width is first optimized using a development data set as usual, to achieve the target accuracy. During this optimization, loose values for the other pruning thresholds are used.

The development data set is then recognized one more time to determine the tightest threshold values that still would retain the final best path. Again loose threshold values for the additional pruning criteria are actually used, but now along each partial path hypothesis the tightest possible pruning thresholds that still would have allowed that path to survive are stored. At each frame these values are updated so that if more loose threshold values were needed to allow the expansion of the path at that frame, the new threshold limits are stored and passed forward to the expanded path hypotheses. At the end of the recognition, the final best path contains for each pruning criterion the tightest possible threshold value with which it would not have been pruned away.

If this method was used to collect only single threshold values from the whole development data set, the obtained values might be overly conservative due to some rare events in the prunings in some particular segments. To obtain more useful thresholds, this analysis is done in small segments (1-3 sentences each). Then from a set of pruning thresholds clear outliers can be removed.

Figure 2 shows an example of a histogram obtained using this procedure when optimizing the threshold for equal word count pruning. There were 581 segments over which the threshold data was collected. In this example, the pruning threshold was set around value 165 so that 99% of the segments were still guaranteed to be recognized with the same accuracy. The global beam width in this case was 210.

5. Evaluation

The effect of the described new pruning criteria were evaluated using two Finnish LVCSR tasks, one speaker dependent and one speaker independent task.

The material for the speaker dependent task was a book read by a professional female speaker, the same as used in [5]. For the training, we used 12 hours of data, the development data set was about 19 minutes and the evaluation set 27 minutes. The development set was divided into 47 segments of two to three sentences for estimating the pruning thresholds.

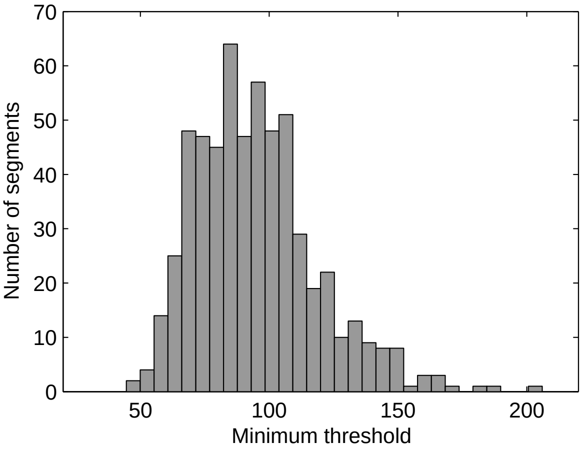

Figure 2: *An example of a pruning threshold measurement.*

For the speaker independent task we used the Finnish SPEECON database [6]. Only adult speakers with clean recording conditions were used. For the training, we took 207 speakers, with total of about 26 hours of material. Both sentences and single words were used for acoustic training. The development and evaluation data sets comprised of disjoint sets of 20 and 31 speakers, respectively. For these two, only read speech sentences without mispronunciations were used. For each speaker, 26–30 of those sentences were available, so that the pruning thresholds could be optimized over 581 segments.

Our speech recognition system uses HMMs and Gaussian mixture models for acoustic modeling. The HMM states of the triphone models have been tied using a decision tree based algorithm [2]. No acoustic adaptation was used. The language model was a sub-word based 4-gram model [5]. The decoder to which the prunings were implemented was our one-pass time synchronous decoder. It uses a static search network for lexicon and acoustic models, and combines the language model dynamically during the search. The decoder is able to model the cross-word triphones properly. A detailed description of the decoder can be found in [7].

Tables 1 and 2 show the recognition results for the two tasks. To better illustrate the effect the pruning criteria have to the search space, the time used for the Gaussian mixture model computations were omitted from the real-time (RT) factors shown in the tables. For the speaker independent task the recognition was run with and without the proposed pruning criteria. With the speaker dependent task we also analyzed the effect of all the proposed criteria separately showing that each of them are useful in restricting the search space. In all the cases the new pruning criteria were used in addition to the basic pruning methods described in Section 2. The global beam width and the histogram pruning threshold were first optimized iteratively to achieve a close to optimal accuracy in the development data set. Then the other pruning criteria were optimized as described in Section 4. The beam widths for global and word end prunings were adapted during the decoding to minimize the need for histogram pruning.

The results show that the proposed pruning criteria can reduce the search space over 50% as in the speaker dependent

Table 1: *Evaluation, speaker dependent task.*

Prunings	Word error rate	Phoneme error rate	RT factor (search)
Only basic	12.6%	1.96%	1.4
Equal depth	12.7%	1.96%	1.1
Equal WC	12.7%	1.95%	1.0
Fan-in	12.6%	1.95%	1.2
All	12.6%	1.95%	0.65

Table 2: *Evaluation, speaker independent task.*

Prunings	Word error rate	Phoneme error rate	RT factor (search)
Only basic	31.3%	10.2%	7.1
All	31.4%	10.3%	5.6

task, without compromising the recognition accuracy. However, the performance in a more difficult task (speaker independent, no adaptation, limited amount of training material) was lower, and the search space computations were reduced only about 20%.

6. Conclusions

This paper presented three new pruning criteria, the equal depth pruning, the equal word count pruning, and the fan-in pruning, which can be used to reduce the search space in decoding. The criteria can be easily implemented to static search tree based decoders, and they are probably applicable in some other decoder architectures as well. The evaluation of the pruning criteria in two LVCSR tasks showed that the reduction of the search space can be over 50%. This reduction, however, is task dependent, and the performance boost obtained in a more difficult speaker independent task was only about 20%. This was probably due to a rather poor overall recognition accuracy (word error rate about 31%), implying significant acoustic confusion with respect to acoustic models.

We also presented a method for easily determining the threshold values for different pruning criteria. This is useful especially when the number of different pruning criteria increases, so that it is no longer feasible to simply try a set of different values for each criterion. The threshold optimization method presented guarantees that some portion of the segments in a development data set will be recognized the same way as with the preset loose threshold values. Even tighter threshold values might still lead to the same recognition results, but with different state or phoneme level segmentations. In practice, however, the values obtained using this method are close to optimal for the selected level of accuracy.

7. Acknowledgments

This work was supported by the Academy of Finland in the project “New adaptive and learning methods in speech recognition”. We thank the Finnish Federation of the Visually Impaired and the Departments of Phonetics and General Linguistics of the University of Helsinki for providing speech data. We also thank the Finnish news agency (STT) and the Finnish IT center for science (CSC) for the text data.

8. References

[1] X. L. Aubert, “An overview of decoding techniques for large vocabulary continuous speech recognition,” *Computer Speech and Language*, vol. 16, pp. 89–114, 2002.

[2] J. J. Odell, “The use of context in large vocabulary speech recognition,” Ph.D. dissertation, Queens’ college, 1995.

[3] S. Ortmanns and H. Ney, “Look-ahead techniques for fast beam search,” *Computer Speech and Language*, vol. 14, pp. 15–32, 2000.

[4] H. Van Hamme and F. Van Aelten, “An adaptive-beam pruning technique for continuous speech recognition,” in *Proceedings of ICSLP*, 1996, pp. 2083–2086.

[5] V. Siivola, T. Hirsimäki, M. Creutz, and M. Kurimo, “Unlimited vocabulary speech recognition based on morphs discovered in an unsupervised manner,” in *Proceedings of Eurospeech*, 2003, pp. 2293–2296.

[6] D. Iskra, B. Grosskopf, K. Marasek, H. van den Heuvel, F. Diehl, and A. Kiessling, “SPEECON - speech databases for consumer devices: Database specification and validation,” in *Proceedings of LREC*, 2002, pp. 329–333.

[7] J. Pylkkönen, “An efficient one-pass decoder for Finnish large vocabulary continuous speech recognition,” in *Proceedings of 2nd Baltic conference on human language technologies*, 2005, pp. 167–172.