Search Space Pruning Based on Anticipated Path Recombination in LVCSR

David Nolden, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany
{nolden, schlueter, ney}@cs.rwth-aachen.de

Abstract

In this paper we introduce a well-motivated abstract pruning criterion for LVCSR decoders based on the anticipated recombination of HMM state alignment paths. We show that several heuristical pruning methods common in dynamic network decoders are approximations of this pruning criterion.

The abstract criterion is too complex to be applied directly in an efficient manner, so we derive approximations which can be applied efficiently.

Our new pruning methods allow much more exhaustive pruning of the search space than previous methods. We show that the size of the search space can be reduced by up to 50% at equal precision over the previous state of the art, and the RTF by 20%.

The abstract pruning criterion can be considered a guide to derive effective pruning methods for any kind of time synchronous decoder.

Index Terms: speech recognition, search, pruning

1. Introduction

Pruning of the search space is critical for efficient state-of-the-art LVCSR decoding. For dynamic network decoders, many advanced pruning heuristics have been developed over time [1] [2] [3].

Recently weighted finite state transducer (WFST) decoders have become a popular approach [4]. However, for WFST decoders typically global beam pruning is the only facilitated pruning method, since there is no strict requirement for further pruning, and the search network is missing intuitive points of application.

In [3] we have shown that some of the advanced pruning methods common in dynamic network decoders, like word end pruning and LM state pruning, are not only tricks to make dynamic network decoders work efficiently, but also well-motivated ways of pruning the search space independently of specific decoder requirements.

In this work we turn the motivations behind advanced pruning described in [3] into a new abstract pruning criterion for state hypotheses, which is independent of the actual decoding architecture, and allows holistic pruning of the whole search space. Since the abstract pruning criterion is too difficult to be evaluated directly, we derive approximations which make its application feasible. We evaluate the resulting pruning methods experimentally on a state of the art dynamic network decoder, and show that the methods allow a significant reduction of the search space and real time factor (RTF) at equal precision.

2. Pruning Based on Anticipated Path Recombination

In the following we assume a minimized single-word search network (our considerations can be applied to WFST decoders by establishing a straightforward mapping).

When two HMM state hypotheses (s_1, q_1) and (s_2, q_2) with network state s_i and probability q_i share a common state $s_1 = s_2$, then all possible following HMM alignment paths produce the same acoustic score (we say that their *paths are recombined*). The only source of further discrimination between recombined paths is the LM, therefore a very tight pruning is possible on such state hypotheses: We call it *LM state pruning* [3], a common pruning technique which is specifically important in token-passing decoders [5].

When the paths of s_1 and s_2 are not yet recombined but can be expected to be recombined within a short interval (for example within 2 timeframes), then only few discriminating acoustic scores can be accumulated before the paths merge, therefore the two state hypotheses can be pruned much sharper than a random pair, albeit not as sharply as with LM state pruning.

2.1. Monotonicity and Convergence

In [3] we have introduced two basic assumptions regarding the flow of state hypotheses during decoding which can be exploited to anticipate the path recombination interval: *Monotonicity* and *convergence*.

The *monotonicity* assumption states that, even though HMMs allow loops, *likely* HMM state alignment paths proceed forwards on the state index axis at a relatively constant rate, depending on the rate of speech in the underlying signal.

The *convergence* assumption states that likey HMM state alignment paths converge while they are aligned with the same state sequence.

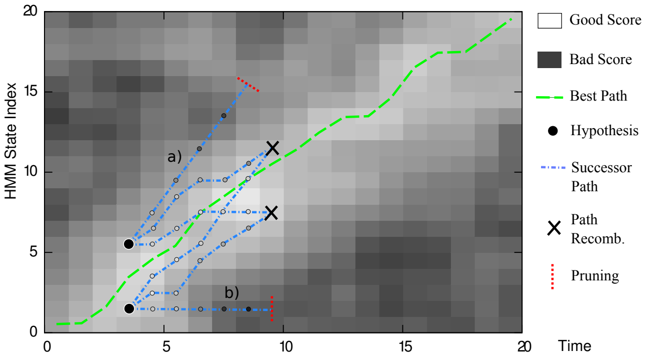


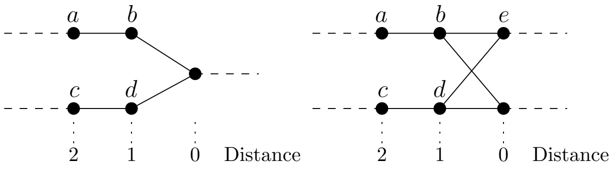
Figure 1: *Convergence and monotonicity.*

Figure 1 illustrates both assumptions. The acoustic model assigns better scores to hypotheses which are somewhat consistent with the speech in the signal, thereby forming a valley around the best path, which all likely paths are pushed towards. Path $b)$ illustrates the monotonicity assumption: It does not progress on the HMM state index axis (eg. it loops) and accumulates bad scores until it is pruned away by global beam pruning. Path $a)$ illustrates the convergence assumption: It progresses too fast, accumulates bad scores, and is then pruned. For each likely successor path behind each of the two illustrated hypotheses, there is a likely path behind the other hypothesis which crosses the path within a short interval.

These assumptions are not strictly true (consider the silence model), however it suffices if they *approximately* are.

2.2. Path Recombination Interval

If we want to prune state hypothesis (s_1, q_1) relative to (s_2, q_2), we need to know the interval until *every* successor path of s_1 is



(a) Bottleneck recombination (b) Cross recombination

Figure 2: *HMM search network recombination examples.*

recombined with *at least one* successor path of s_2. We call this the *asymmetric path recombination interval*.

Based on the monotonicity and convergence assumptions we can define a simple approximation of the anticipated asymmetric path recombination interval:

$$r(s_2 \rightarrow s_1) := \min\{d_1, d_2\} + f_{CONV} \cdot |d_1 - d_2| \quad (1)$$

Where d_1 is the shortest distance behind s_1 at which all paths following s_1 intersect at least one path following s_2, and d_2 is the corresponding distance behind s_2. f_{CONV} is a parameter defining the rate of convergence, a low value indicates fast convergence, infinite indicates no convergence at all.

We ignore the rate of speech, as that would be a linear factor, redundant with pruning parameters we will define later.

The chosen approximation is consistent with our monotonicity and convergence assumptions: Due to monotonicity the interval is linear in the distances d_1 and d_2, and due to convergence asymmetric hypotheses converge by the factor f_{CONV}.

Figure 2a demonstrates simple linear path recombination in a HMM search network. All followup paths behind states a and c are symmetrically recombined at distances of $d_1 = d_2 = 2$, which leads to a recombination interval of $r(c \rightarrow a) = 2$. The pair a and d is not symmetric, so the convergence factor f_{CONV} shows an effect, and the interval is $r(d \rightarrow a) = 1 + f_{CONV}$.

In real search networks the followup paths of states typically don't run together on a single path, but on multiple paths in parallel, as shown in Figure 2b. This makes it more difficult to identify the point of intersection, but the network mostly produces the same intervals as the network in Figure 2a.

The path recombination interval is highly asymmetric: In the network shown in Figure 2b, *one* path starting at b intersects *all* paths starting at e, and therefore $r(b \rightarrow e) = f_{CONV}$. However only one of the two paths starting at b is intersected by paths starting at e, therefore the recombination interval $r(e \rightarrow b)$ is undefined, depending on the structure of the hidden rest of the network.

2.3. Pruning Criterion

We define our abstract pruning criterion based on anticipated path recombination as follows:

$$\text{Discard } (s_1, q_1) \text{ if } q_1 < q_2 \cdot f_L \cdot f_A^{r(s_2 \rightarrow s_1)} \quad (2)$$

Where (s_2, q_2) is any other hypothesis, f_L is the pruning threshold applied at a recombination interval of zero (eg. the LM state pruning threshold), and f_A is the factor by which the recombination interval affects the sharpness of pruning.

This pruning criterion is consistent with our motivations of pruning: When the successor paths are already recombined (eg. the recombination interval is zero), then LM state pruning is applied, with higher recombination intervals the threshold is scaled according to the number of discrimating scores which can be accumulated until the paths merge.

2.4. Practical Feasibility

It is very difficult to apply the defined pruning criterion directly: Recombination can occur at many different spots in the search network, the recombination distances d_1 and d_2 for pairs of states are difficult to compute, and it is not feasible to consider each pair of active state hypotheses, since that imposes a quadratic runtime effort. Approximations are required to apply the pruning criterion efficiently.

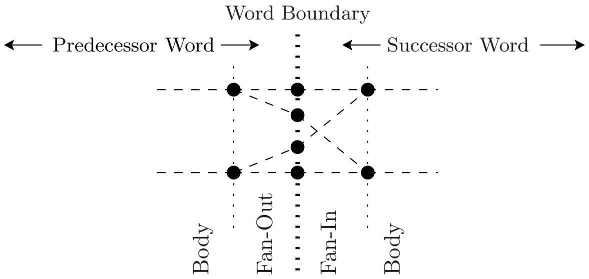


Figure 3: *Simplified recombination paths at word boundaries in common LVCSR search networks with across word modelling.*

3. Approximations

In modern LVCSR decoders most of the path recombination happens at word boundaries, therefore it is possible to gain a large portion of the potential effect by focusing on that specific point of recombination.

3.1. Symmetric Recombination Interval

Figure 3 shows a simplification of the recombination paths at word boundaries in common LVCSR decoders with across-word modelling. Due to across-word modelling, paths belonging to the predecessor word split up already in the fan-out before the physical word boundary, according to the right acoustic context. In the successor word's fan-in, every path originating from the predecessor word's body crosses at least one path originating from any other portion of the predecessor word's body, because each word can be followed by *any* other word.

We define the states at the beginning of the successor word's body to be the *recombination line*. We can focus on the recombination line as a pessimistic upper bound for path recombination of states belonging to the predecessor word's body (see Figure 4), because all paths following the predecessor body cross before that line.

To focus on the recombination line, we define the recombination interval symmetrically:

$$r(s_1 \leftrightarrow s_2) := \min\{d(s_1), d(s_2)\} + f_{CONV} \cdot |d(s_1) - d(s_2)| \quad (3)$$

Where $d(s)$ is the shortest distance at which every successor path of state s reaches the recombination line. Equation 3 is a valid symmetric approximation of Equation 1, as long as s_1 and s_2 are in the body of the predecessor word (see Figure 4).

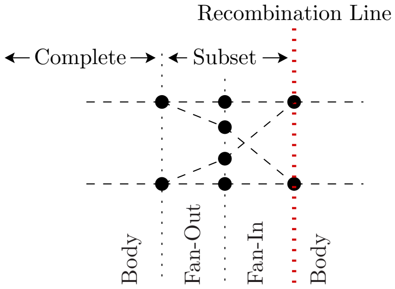


Figure 4: *Reachability of the recombination line. The complete recombination line is reachable from the predecessor word's body, only a subset is reachable from fan-out and fan-in.*

3.2. Asymmetric Body Pruning

Pruning based on our criterion and the symmetric recombination interval can be implemented efficiently: The recombination interval $r(s_1 \leftrightarrow s_2)$ only depends on the recombination line distances $d(s_1)$ and $d(s_2)$, so pruning thresholds can be precomputed for each distance, and the main effort can be reduced to pairs of distances, rather than pairs of hypotheses.

The pruning can be broken down to 3 steps:

1. For each recombination line distance d_1, collect the highest state hypothesis probability:
$$q(d_1) := \max_{(s,q),d(s)=d_1} q \tag{4}$$
2. Compute the relative pruning thresholds:
$$p(d_1) := \min_{d_2 \geq B} q(d_2) \cdot f_L \cdot f_A^{r(d_2 \leftrightarrow d_1)} \tag{5}$$
Where B is the combined length of fan-out and fan-in.
3. Prune:
$$\text{Discard } (s_1, q_1) \text{ if } q_1 < p(d(s_1)) \tag{6}$$

The main effort are the runs over all state hypotheses in step 1 and 3, which can be integrated into the standard beam pruning.

Since only states belonging to the predecessor word's body reach the complete recombination line (see Figure 4), only those are considered as sources of pruning, by computing pruning thresholds relative to distances $d_2 > B$.

Each word can appear both at the left and right side of the word boundary. Therefore, in step 3, each state s is pruned twice with alternative distances $d(s)$: Once interpreting the state as a part of the predecessor word (eg. with high distances $d(s)$), and once interpreting the state as a part of the successor word (eg. with low or even negative distances $d(s)$, see Figures 3 and 4). When pruning hypotheses in the successor word interpretation, we apply an additional factor f_{DIS} to the pruning threshold to compensate potential discontinuities of LM scores at word boundaries.

3.3. Recombination Set Pruning

Asymmetric body pruning only allows pruning relative to the predecessor word's body. However, a large portion of the active search space typically belongs to the fan-out or fan-in, because there the branching factor is very high.

The condition for pruning s_1 relative to s_2 is: Every successor path of s_1 must cross *at least one* successor path of s_2 (see Subsection 2.2). This condition can be verified for states belonging to the fan-out and fan-in by comparing the exact subset of the recombination line which is reachable from those states.

We define the *recombination set* to be the subset $u(s)$ of the recombination line reachable from state s (see Figure 4).

We can prune state s_1 relative to s_2 if $u(s_1) \in u(s_2)$. For states s_2 which are part of the predecessor word's body, where $u(s_2)$ equals the complete recombination line, this results in the same pruning criterion as asymmetric body pruning. However, we can now prune states in the fan-in and fan-out relative to other states from the fan-in and fan-out, based on the intersection of their recombination sets.

Recombination sets can be collected in a preprocessing step, and the number of different sets is limited when the structure of the network follows the pattern illustrated in Figure 3.

To apply the pruning efficiently, once again it is not feasible to consider individual state hypothesis pairs. Recombination set pruning can be implemented somewhat efficiently by prepartitioning all computed sets according to their subset relationships, collecting the relevant state hypothesis information for each recombination set similar to the algorithm from Subsection 3.2, and propagating distance-dependent pruning thresholds from the supersets into the subsets. The collection of state hypothesis information and the actual pruning is very efficient, but the propagation of pruning thresholds between the recombination sets imposes an effort linear their absolute number.

3.4. Fan-Out Short Word Pruning

The runtime overhead of recombination set pruning is significant. The effort is justifiable on relatively slow setups, however under many conditions the overhead may be too high.

In many languages, the most common words are typically very short (think of “a”, “the”, “an”, “in”, “of”, etc.).

Fan-out states will often be followed by a short word in the fan-in structure. By definition, the short word can be followed by any other word, therefore the complete recombination line is reachable from the fan-out states *through* that short word with a delay equal to the length of the word.

We can exploit this delayed reachability of the recombination line by relaxing the constraint B while computing pruning thresholds (see Equation 5) by the length of the fan-out, and applying an additional factor f_{SW} to the threshold to compensate the delay and the loss in precision due to the approximation.

4. Relation to Common Pruning Methods

Word end pruning [3] is an approximation of our pruning criterion, because word ends usually have an equal recombination line distance D_{WE} in dynamic network decoders. Therefore the word end pruning threshold can be derived directly: $f_{WE} = f_L \cdot f_A^{D_{WE}}$. If word end labels are placed in the fan-out rather than the body, then the motivation of short word pruning can be applied (see Subsection 3.4).

Word end pruning fade-in [3] is a more complete approximation, and equals our body pruning with $f_{CONV} = \infty$.

5. Experiments

We use a dynamic network decoder based on the word conditioned approach with a partially minimized search network [5]. The acoustic scores are computed efficiently using quantized features, temporal batching, and Gaussian preselection. Model-based and temporal acoustic look-ahead [6] and sparse full-order 4-gram LM look-ahead [7] are used to focus the search.

We perform our experiments on the first speaker-independent pass of the RWTH Aachen Quaero English ASR system [8]. The lexicon comprises 158k words with 180k pronunciations, modeled by 45 phonemes and 6 non-speech phones, and the 4-gram LM is composed of 50M n-grams. The acoustic model comprises 4501 Gaussian mixture models with a globally tied covariance matrix and 1M mixture densities. The test corpus consists of 1482 segments with a duration of 3.4h and about 36k spoken words.

Real time factors (RTF) are measured on a 24-core AMD Opteron 6176 machine with 2.3Ghz and 64GB of memory (without parallelization).

During our experiments we have noticed that non-approximated full-order LM look-ahead is crucial for our pruning methods to work, because only then the LM scores are distributed smoothly over the search network. Significant discontinuities in the LM scores break the correlation between the anticipated recombination interval and the actual scores, and thereby would render our pruning criterion useless. Simple tricks like computing LM look-ahead tables with some delay can not be facilitated. We were able to compensate the runtime overhead caused by disabling such tricks through more aggressive caching of look-ahead tables.

5.1. Results

Our new pruning methods are difficult to tune, because there are many inter-dependent parameters: The convergence factor f_{CONV}, the LM state pruning threshold f_L, the recombination interval factor f_A and the discontinuity factor f_{DIS}.

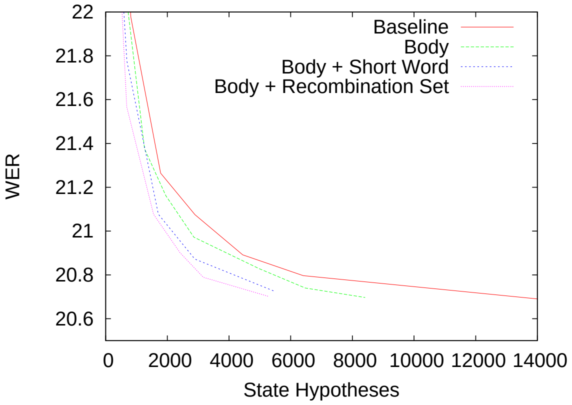


Figure 5: *WER vs. average active state hypotheses.*

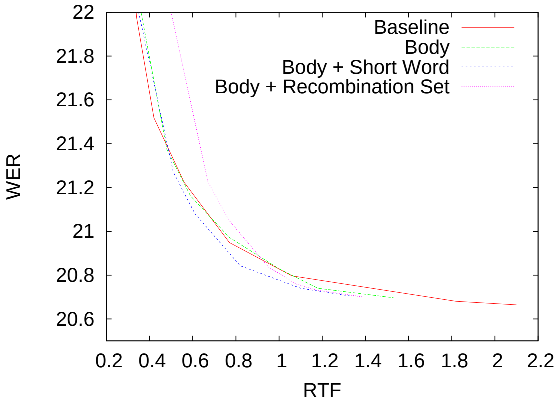


Figure 6: *WER vs. RTF.*

We use an ideal set of combinations between global beam pruning and word end pruning thresholds as baseline.

To tune the new parameters, we have performed a large grid of recognitions on a dev set with varying values for all new parameters, and ultimately selected a flattened pareto frontier.

For f_{CONV}, f_{DIS} and f_A the optimization selected globally consistent optima: $f_{CONV} = 4$, $f_{DIS} = 40$ and $f_A = 6$ (illustated by the negative logarithm). The optimal value of f_L scales with the global beam pruning threshold. For short word pruning, an optimal factor of $f_{SW} = 80$ was selected. For recombination set pruning, 5k sets were observed, each with approximately 3 direct sub-sets.

Figure 5 shows the optimal relationships between WER and the size of the search space. Body pruning reduces the size of the search space by approximately 20% at equal precision. Recombination set pruning reduces the search space by approximately 50% at equal precision. Short word pruning reduces the search space by approximately 40%, and comes quite close to the recombination set pruning. Combining recombination set pruning with short word pruning brings no gain at all, which indicates that short word pruning is a direct approximation of recombination set pruning.

Figure 6 shows the relationship between WER and RTF. Body pruning improves the relationship between WER and RTF when close to the optimal WER. At higher WERs, when the search space is smaller, the effort of the additional pruning step seems to cancel the gain from the reduced search space. Recombination set pruning has a significant static runtime overhead of about 0.15 RTF, so it is not competitive for the higher error rates and faster configurations. Only very close to the best WER the static overhead starts paying out and recombination set pruning becomes competitive. Short word pruning allows nearly the same reduction in search space as recombination set pruning, while it induces no runtime overhead at all compared to body pruning, therefore short word pruning is the best performing pruning method regarding RTF, leading to a reduction of the RTF by approximately 20% at equal precision.

We have verified the results on the more difficult quaero Polish task with a 600k word vocabulary, and obtained similar results. Despite the very different task, the optimal parameters f_{CONV} f_A and f_{DIS} were selected equally on quaero Polish, only for f_L different optima were observed, which indicates that this is the only parameter which is task-dependent.

6. Conclusions and Outlook

Our new pruning criterion allows reducing the search space by 50% and the RTF by 20% at equal precision on our state of the art dynamic network decoder.

Application to WFST decoders is very tempting, because a more significant correlation between the size of the search space and the efficiency can be expected there, LM scores are inherently distributed smoothly over the whole search network due to weight pushing (which is a critical condition for the new pruning criterion to perform well), and there is a general lack of advanced pruning methods for WFST decoders.

Our experiments indicate that the currently predominant approach of maximal minimization and determinization does not yield the perfect search network regarding pruning. To optimize the pruning based on anticipated path recombination, the search network should be transformed so that the majority of active state hypotheses are as close as possible to the recombination line, which would mean a partial nondeterminization of the search network.

7. Acknowledgements

This work was partly realized under the Quaero Programme, funded by OSEO, French State agency for innovation.

8. References

[1] H. Soltau and G. Saon, “Dynamic Network Decoding Revisited,” in *ASRU*, 2009.

[2] H. Ney and S. Ortmanns, “Progress in Dynamic Programming Search for LVCSR,” in *Proceedings of the IEEE*, vol. 88, no. 8, Barcelona, Spain, August 2000, pp. 1224 – 1240.

[3] D. Nolden, R. Schlüter, and H. Ney, “Extended search space pruning in lvcsr.” ICASSP, 2012.

[4] M. Mohri, F. Pereira, and M. Riley, “Speech Recognition with Weighted Finite State Transducers,” in *Handbook of Speech Processing*. Springer, 2008, pp. 559–582.

[5] D. Nolden, D. Rybach, R. Schlüter, and H. Ney, “Joining advantages of word-conditioned and token-passing decoding.” ICASSP, 2012.

[6] D. Nolden, R. Schlüter, and H. Ney, “Acoustic Look-Ahead for More Efficient Decoding in LVCSR,” in *Interspeech*, Florence, Italy, August 2011.

[7] D. Nolden, H. Ney, and R. Schlüter, “Exploiting Sparseness of Backing-Off Language Models for Efficient Look-Ahead in LVCSR,” in *ICASSP*, Prague, Czech Republic, May 2011.

[8] M. Sundermeyer, M. Nußbaum-Thom, S. Wiesler, C. Plahl, A. El-Desoky Mousa, S. Hahn, D. Nolden, R. Schlüter, and H. Ney, “The RWTH 2010 Quaero ASR Evaluation System for English, French, and German,” in *ICASSP*, Prague, Czech Republic, May 2011, pp. 2212–2215.