EXTENDED SEARCH SPACE PRUNING IN LVCSR

David Nolden, Ralf Schlüter, Hermann Ney
Chair of Computer Science 6, RWTH Aachen University
Ahornstr. 55
D-52056 Aachen, Germany
{nolden, schlueter, ney}@cs.rwth-aachen.de

ABSTRACT

We compare the most important pruning methods which are common in different LVCSR decoding architectures and lead them back to a theoretical motivation. Based on this motivation, we propose a new pruning method which fades the word end pruning over a large part of the search network. We analyze the methods regarding their relationship between search-space and word error rate, and regarding their mutual dependence.

We show that the different pruning methods are mutually dependent and difficult to combine, and that our new pruning method is the most effective method regarding both the search space and runtime efficiency.

Index Terms— LVCSR, search, decoding, word conditioned, tree-search, pruning

1. INTRODUCTION

In dynamic network decoders, the language model (LM) and acoustic model (AM) are combined dynamically. The acoustic model is used to build a compact HMM search network representing all the words in the vocabulary, and the LM dependencies are maintained by appropriate dynamic management of state hypotheses [1].

Static decoders on the other hand, usually based on the weighted finite state transducer (WFST) approach [2], combine the AM and LM statically by building one huge HMM search network integrating both models.

In all common decoding architectures, *acoustic pruning* is used to keep the effort tractable, by propagating state hypotheses through the search network time-synchronously, and applying a global *beam* to the state hypotheses, discarding state hypotheses which are worse than the best one by a certain threshold.

It is beneficial to exploit as much *future* knowledge as possible to focus the search: With *LM look-ahead* the probabilities reachable word ends are included during pruning [3], and with *acoustic look-ahead*, the expectation regarding future acoustic observations is integrated [4].

In WFST decoders, acoustic pruning is typically the only applied pruning method. In dynamic network decoders with a static single-word search network on the other hand, certain additional pruning methods are required. A critical pruning method is the *word end pruning*, which reduces the effort required for word end handling, one of the inherent bottlenecks in dynamic network decoders (LM probabilities need to be computed, look-ahead structures generated, etc.).

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

In token passing decoders [5], where lists of LM tokens are attached to the states of the search network, *LM state pruning* is a critical pruning method reducing the number of tokens assigned to each state.

In [6] three additional pruning methods for dynamic network decoders were proposed, however they miss a proper motivation, and they were not analyzed thoroughly, because they were not compared with the acoustic pruning as baseline pruning method.

In this work, we try to motivate the different pruning methods consistently. We analyze and compare the different methods experimentally, and we propose a new pruning method which is consistent with our expectation about the flow of state hypotheses through the search network.

2. DECODER

Our dynamic network decoder is based on the word conditioned search (WCS) architecture [1], extended to minimize the suffix of the search network [7]. Combined with *LM state pruning*, the decoder can represent both WCS and token-passing decoders regarding the search space, depending on the chosen pruning thresholds. The whole one-word HMM search network is expanded statically.

The search network can be split into three parts: A minimized fan-in which models the across-word coarticulation at word starts, a tree-like body following the fan-in where paths split up, and a minimized fan-out behind the body modelling the across-word coarticulation at word ends.

The active search space is managed dynamically by propagating state hypotheses through the search network time-synchronously and handling word ends and path recombination according to the Viterbi approximation. Each state hypothesis (s, h, q) stands for a path ending in state s of the network with the LM history h, and partial path probability q.

3. STANDARD PRUNING

The most important pruning method is the global *acoustic pruning*: At each timeframe, all state hypotheses which have a lower probability than the best one multiplied by a specific pruning threshold are discarded.

Whenever a word end label is encountered during decoding, a word end hypothesis is created and the followup root state with extended LM history is activated. Since new LM look-ahead tables need to be initialized for each new unique LM history, it is very important to keep the number of word end hypotheses low. Therefore word end hypotheses are pruned at each timeframe by discarding all word end hypotheses which have a probability lower than the best one multiplied by a specific pruning threshold [1] (we call this *word end pruning*).

Additionally *histogram pruning* is used to limit the absolute number of state- and word-end hypotheses that appear at each timeframe. Histogram pruning is required to cut off peaks in situations of high uncertainty, and does not play a significant role regarding the runtime efficiency.

We have shown in [8] that preventing transitions into word starts at a specific fraction of all timeframes does not increase the WER. We transfer this concept into the word conditioned decoder by only handling word ends at each *i*th timeframe (we call this *word end interval*).

3.1. LM State Pruning

LM state pruning is common in token-passing decoders, because in token-passing decoders the number of state hypotheses (s, h, q) that are active on a single state s is critical for efficient recombination.

The state hypothesis (s, h, q) is removed if there is another state hypothesis (s, h', q') on the same network state s with probability q' better by a specific threshold.

If two state hypotheses share a state $s = s'$ in the search network, the relative probabilities of all followup paths through the network leading to a sentence-end can only be discriminated by the LM (the AM assigns equal probabilites to equal HMM state alignments). The AM has a much stronger influence on the overall hypothesis probabilities than the LM, thus a majority of the variability that can discriminate the followup hypotheses has fallen away. Therefore the LM state pruning threshold can typically be much tighter than the acoustic pruning threshold without introducing additional errors.

LM state pruning is especially effective if the static search network has a minimized fan-out (see Section 2), because paths that were split up in the body of the network start intersecting in the minimized fan-out, and may be pruned before reaching the fan-in.

4. FOUNDATIONS OF PRUNING

LM state pruning is motivated well because the LM is the only source of discrimination between the affected hypotheses.

Other pruning methods like the word end pruning are very effective too though, although the pruned word end hypotheses do not have much in common (both the AM as well as LM context are different).

We will introduce two assumptions regarding the flow of state hypotheses during decoding, which will help us explaining why word end pruning works, and which will motivate our new pruning method.

4.1. Monotonicity

The HMMs used for speech recognition typically allow loops as well as skips, so many extremely different HMM state alignments are possible. However the acoustic models assign higher probabilities to paths that propagate through the search network monotonously at a speed which depends on the actual speed of the speech in the acoustic signal. Since acoustic pruning removes the less likely hypotheses, we can expect all the followup state hypotheses of one origin state hypothesis to propagate *monotonously* away into the depth of the search network, and we can assume that for each distance d, we can define an interval Δt after which all followup state hypotheses have taken at least d forward transitions away from the origin. For example, it is very unlikely that the same state hypothesis is kept alive for many timeframes by repetitious HMM loops. The only exception are non-speech models like silence or noise, which tend to stay likely over many consecutive timeframes, due to their stationarity.

4.2. Convergence

The acoustic model is meant to assign the highest probabilities to HMM state alignments which equal the actual speech in the acoustic signal, thus acoustic pruning forces the active state hypotheses to converge towards the parts of the search network which model words and phonemes expressed in the acoustic signal.

Metaphor: The acoustic signal represents a *valley*, which the acoustic pruning as a *gravity* forces all state hypotheses to converge towards.

For each pair of origin state hypotheses (s, h) and (s', h'), we can expect that after a specific interval Δt, all of their follow-up hypotheses have either been pruned away, or end up on the same states of the search network (which the acoustic pruning forced them, with time, to converge towards).

The interval Δt can be expected to be very short when the states are very close to each other (for example, s' is the only direct successor of s), and much larger when they are very distant in the search network.

The described convergence is exploited in the *Word Pair Approximation* technique for lattice generation [9].

5. MOTIVATION OF WORD END PRUNING

Based on the monotonicity and convergence assumption, word end pruning can be motivated as an extension of LM state pruning.

Word labels are placed right before the fan-out (see Section 2). In the fan-out and fan-in structure of the network, each path inherently crosses each other path (because each word can be followed by any other word). Following the monotonicity and convergence assumptions, the followup state hypotheses of different word labels will converge onto the same states of the search network within a relatively short interval. From that point on, the LM will be the only source of discrimination between them, equivalently to LM state pruning (see Subsection 3.1).

Since the followup paths behind word ends merge within a short interval, less discrimination from the AM can be expected before the paths merge than in the general case. Therefore, the word end pruning can typically be much tighter than the acoustic pruning, but less tight than the LM state pruning.

6. NEW METHOD: WORD END PRUNING FADE-IN

The motivation of word end pruning (see Section 5) equally applies to states which are more distant from the fan-out than the word labels themselves.

For example, the successor paths behind two state hypotheses which are each only one forward-transition away from their most distant following word labels can be expected to enter the fan-out very soon (due to monotonicity, see 4.1), and will converge within an interval only slightly longer than for paths behind word labels, therefore a relaxed word end pruning can be applied anticipatively.

Let $d(s)$ be the distance from HMM state s to the most distant following word label in the search network (the distance after which every possible successor path will enter

the fan-out). To fade the word end pruning into the search network, we prune all state hypotheses s with equal word-end distance $d(s)$ using distance-dependent pruning threshold $f_{FD}(d(s))$.

We define the distance-dependent pruning treshold so that it is equal to the standard word end pruning threshold f_{WE} directly on word ends ($d = 0$), and fades to the acoustic pruning threshold f_{AC} over a tunable depth of d_{WE}.

$$f_{FD}(d) = \begin{cases} f_{WE} & d = 0 \\ f_{WE}^{1-d/d_{WE}} \cdot f_{AC}^{d/d_{WE}} & \\ f_{AC} & d > d_{WE} \end{cases} \quad (1)$$

The pruning can only have an influence as long as the threshold is lower than the acoustic pruning threshold, thus the influence fades out over the distance d_{WE}.

7. EXPERIMENTAL RESULTS

We analyze the different pruning methods on 2.85h of parliamentary speech by changing speakers using a LVCSR system tailored for English european union parliamentary speech [10].

The lexicon consists of 53k words and 59k pronunciations, modeled by 45 phonemes and 6 non-speech phones. The acoustic model consists of 4.5k Gaussian mixture models with a globally tied covariance matrix and overall 900k mixture densities. The mixture models are assigned to triphone states using a CART tree. Each triphone is modeled by 3 HMM states with 2 state repititions. The used 4-gram LM contains 7.4M n-grams.

Efficient full-order 4-gram LM look-ahead [3] and acoustic look-ahead [4] are used in conjunction with all tested pruning methods. The RTFs are measured on an AMD Opteron 248 with 2.2 GHz and 8 GB of memory.

8. INDIVIDUAL METHODS

An additional pruning method is only useful if it brings a gain over the standard acoustic pruning, achieving an improved relation between RTF and WER. We compute the optimal curve for each additional pruning method by testing a large number of relevant combinations between acoustic pruning threshold and the additional pruning threshold, and selecting a flattened pareto-frontier regarding WER and RTF. This gives us, for each additional pruning method, a list of threshold-combinations which have the best relationship between WER and RTF. Typically, a tigher acoustic pruning is tied to a tighter additional pruning.

We do not use a separate dev-corpus to tune the thresholds, because we only want to analyze the *potential* effect of the methods given that the thresholds can be perfectly tuned.

An ideal fade-in distance of 50 was selected for the word end pruning with fade-in.

Figure 1 compares the different pruning methods regarding their ideal relation between WER and the size of the search space. For the baseline only the acoustic pruning threshold was varied. Since our decoder requires word end pruning in order to perform well, even the baseline uses word end pruning, however using a fixed suboptimal pruning threshold. For the additional pruning methods, ideal combinations with the acoustic-pruning threshold were selected in the way described above. Surprisingly, even the word end interval leads to a better relationship between search space and WER, although it is not really consistent with our pruning motivation. Ideal word end pruning performs only slightly better than the word end interval, reaching a search space reduction of around 5% at equal error rate. LM state pruning performs very well, reaching a reduction of 10 to 20%. Word end pruning fade-in achieves a reduction of 10 to 20% over word end pruning alone.

We have analyzed the pruning methods proposed in [6], but none of them led to any measurable gain, which is consistent with our motivation of pruning.

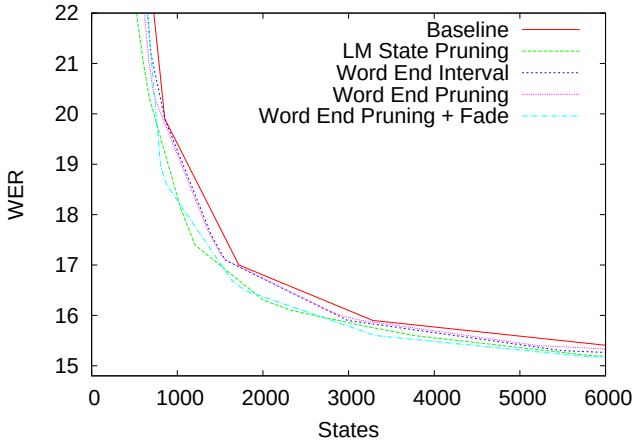

Fig. 1. WER vs. average active state hypotheses.

Figure 2 compares the methods regarding the relation between WER and RTF. LM state pruning performs worse than word end pruning regarding the RTF, because word end pruning directly reduces the effort required for word end handling, which is a critical part of a dynamic decoder, since LM scores need to be calculated, tracebacks managed, etc. The LM state pruning, while it had a stronger influence regarding the search space, has a much weaker effect than the word end pruning, but the reduction is still significant. The novel word end pruning fade-in performs best.

9. METHOD COMBINATIONS

The best performing method alone is the word end pruning with fade-in. Since this includes word-end pruning, the remaining question is how well it can be combined with LM state pruning and with the word end interval. Therefore we proceed similarly to the previous experiments. We use the ideally tuned word end pruning with fade-in as baseline, and combine this baseline with many different thresholds for the additional pruning methods, finally selecting a flatted pareto frontier for each method, yielding the combinations with best relationship between WER and RTF.

Figure 3 shows that, by adding LM state pruning over a well-tuned word end pruning with fade-in, only a very slight improvement in the relationships between search space and WER can be achieved. The improvement achieved by the word end interval is also much lower than without tuned word

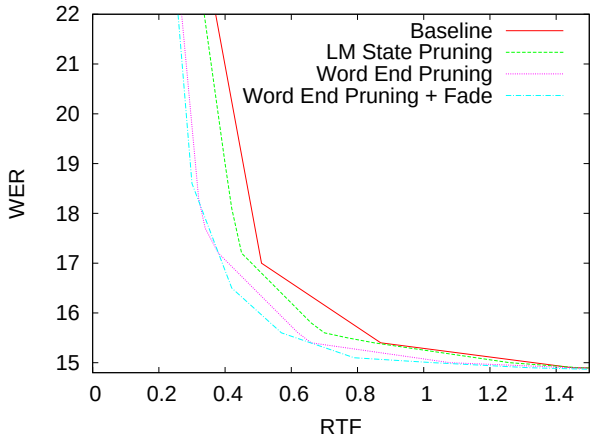

Fig. 2. WER vs. RTF.

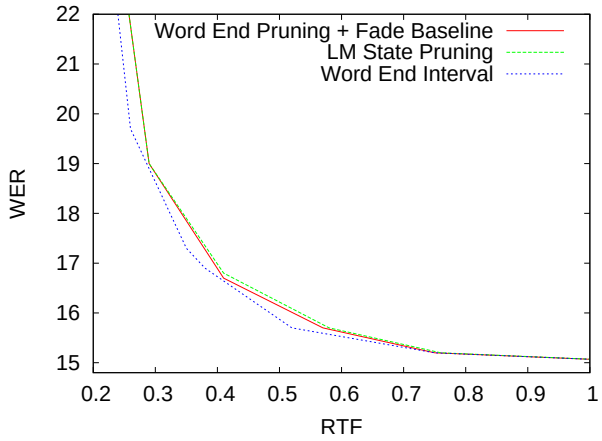

Fig. 4. WER vs. RTF with word end pruning + fade-in as baseline.

end pruning + fade-in, because much less word ends are encountered now.

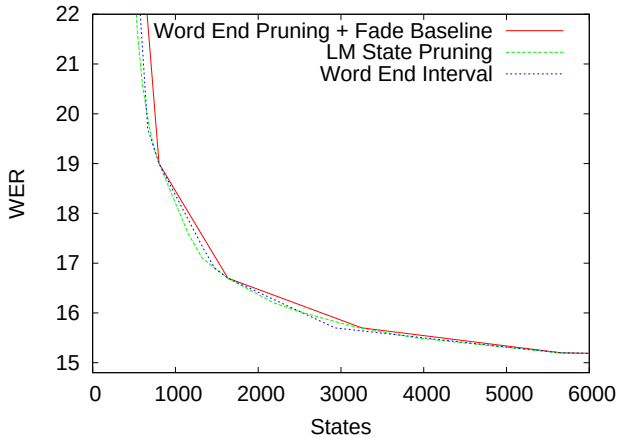

Fig. 3. WER vs. active state hypotheses with word end pruning + fade-in as baseline.

Figure 4 shows that the slight improvement achieved in search space size through LM state pruning does not transfer over to the RTF: The overhead of the additional pruning step is larger than the reduction in runtime achieved through the reduced search space. For each state hypothesis removed through LM state pruning, there stays one other state hypothesis in the same network state, which means that LM state pruning can not reduce the number of acoustic distance calculations, because no acoustic paths are cut off.

The word end interval does show a slight positive effect, although lower than before (see Figure 2).

10. CONCLUSION

It is possible to exploit knowledge about the structure of the search network, and about the flow of hypotheses through the network, to effectively prune the search space.

The word end pruning commonly used in dynamic decoders is not only an implementation detail required for efficient runtime, but also a well-founded way to prune the search space.

By fading the word end pruning over the whole search network as proposed in this work, the search space and RTF can be significantly reduced at equal WER.

11. REFERENCES

[1] H. Ney and S. Ortmanns, “Progress in dynamic programming search for lvcsr,” vol. 88, no. 8. Barcelona, Spain: Proceedings of the IEEE, August 2000, pp. 1224 – 1240.

[2] C. Allauzen, M. Mohri, M. Riley, and B. Roark, “A generalized construction of integrated speech recognition transducers.” Merano: ICASSP, December 2009, pp. 276 – 281.

[3] 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.

[4] D. Nolden, R. Schlüter, and H. Ney, “Acoustic look-ahead for more efficient decoding in lvcsr,” in *Interspeech*, Florence, Italy, August 2011.

[5] S. Young, “A review of large-vocabulary continuous-speech recognition.” IEEE Signal Processing Magazine, September 1996, pp. 45 – 57.

[6] J. Pylkoenen, “New pruning criteria for efficient decoding.” Interspeech, 2005, pp. 581 – 584.

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

[8] D. Nolden, H. Ney, and R. Schlüter, “Time conditioned search in automatic speech recognition reconsidered.” Makuhari, Japan: Interspeech, Sep. 2010.

[9] S. Ortmanns, H. Ney, and X. Aubert, “A word graph algorithm for large vocabulary continuous speech recognition,” vol. 11, no. 1. Computer, Speech and Language, January 1997, pp. 43 – 72.

[10] J. Lööf, M. Bisani, C. Gollan, G. Heigold, B. Hoffmeister, C. Plahl, R. Schlüter, , and H. Ney, “The 2006 rwth parliamentary speeches transcription system.” Barcelona, Spain: TC-STAR Workshop on Speech-to-Speech Translation, 2006, pp. 133 – 138.