DYNAMIC SEARCH-SPACE PRUNING FOR TIME-CONSTRAINED SPEECH RECOGNITION

Sascha Wendt, Gernot A. Fink, Franz Kummert

Faculty of Technology, Bielefeld University,
P.O. Box 100131, 33501 Bielefeld, Germany
{swendt,gernot,franz}@techfak.uni-bielefeld.de

ABSTRACT

In automatic speech recognition complex state spaces are searched during the recognition process. By limiting these search spaces the computation time can be reduced, but unfortunately the recognition rate mostly decreases, too. However, especially for time-critical recognition tasks a search-space pruning is necessary. Therefore, we developed a dynamic mechanism to optimize the pruning parameters for time-constrained recognition tasks, e.g. speech recognition for robotic systems, in respect to word accuracy and computation time. With this mechanism an automatic speech recognition system can process speech signals with an approximately constant processing rate. Compared to a system without such a dynamic mechanism and the same time available for computation, the variance of the processing rate is decreased greatly without a significant loss of word accuracy. Furthermore, the extended system can be sped up to real-time processing, if desired or necessary.

1. INTRODUCTION

Automatic speech recognition (ASR) is a fundamental component for computer-based systems, that are to be instructed by humans. The performance of the speech recognizer should primarily be acceptable for the human. But for time-critical applications involving interaction with manipulation systems, mobile robots, or virtual-reality environments, there is a second demand: the reaction time for spoken commands should be as fast as possible in order to avoid unnatural time-delays or even damage. Therefore, in critical situations the recognizer should be able to process speech input in real-time - at least approximately.

Fast speech recognition can be achieved by different approaches. One possibility is to develop efficient algorithms, which can speed up the system considerably. But by this approach it is not possible to guarantee a worst case performance, if the search space changes dynamically. However, fast speech recognition can also be achieved by limiting the state spaces, that are searched during the recognition process. In the literature several pruning techniques are proposed. Unfortunately, the recognition rate always decreases, if the search space is pruned. Thus, an optimal pruning (pruning with loss of recognition accuracy as small as possible) under certain time-constraints can only be achieved by an optimization process. A second reason for such a process is, that most pruning techniques can't guarantee a certain processing time by themselves. But if the search-space pruning is controlled dynamically by an optimization process, it should also be possible to guarantee a worst case performance - at least within certain limits.

Therefore, we developed a control mechanism for dynamic search-space pruning that allows time-constrained speech recognition. It optimizes different pruning parameters for an ASR system, that uses beam pruning at different levels of a time-synchronous search process. Time-constraints imposed on the recognition process can be specified by the real-time factor (RTF), that is defined as:

$$\mathrm{RTF} = \frac{\text{computation time}}{\text{length of processed speech}} \tag{1}$$

By measuring the current RTF during the recognition process it can be decided if the preset time-constraints are met or the system needs to speed up processing.

2. SEARCH-SPACE PRUNING TECHNIQUES

In the literature many different pruning techniques have been proposed. Some of them will be described shortly in this section.

The most widely used pruning method is defined by the *beam search* algorithm [1]. Only hypotheses with scores close to the currently best solution are retained for further consideration. The width of this score interval is controlled by a single parameter, that needs to be chosen heuristically and is normally kept fixed for the whole search process. Originally designed for limiting the acoustic search space this pruning method can also be applied at the level of mixture density evaluation or language model search (cf. e.g. [2]). However, this pruning technique can't guarantee a certain processing time. If in the worst case all scores lie very close to each other hardly any hypothesis can be rejected and the method degrades to full search.

A similar intention as beam search pruning has *histogram pruning*, which limits the number of surviving state hypothesis to a maximum [3]. In contrast to beam pruning, this technique can guarantee at least a certain worst case processing time. Even if all scores are very close to each other, only the preset maximum number of hypothesis will survive.

In both beam search and histogram pruning a loss of recognition accuracy is to be expected, because the optimal solution can be lost by pruning. For this reason, several *look-ahead* techniques have been proposed to improve search space pruning (e.g. in [4, 5]). These methods also speed up conventional beam search as the processing can be focused even more tightly on promising parts of the search space.

Another pruning technique is time-conditioned merging of search trees. An example for this technique is *merging of adjacent search trees* [6]. Trees ending at time Δt with the same state

In International Conference on Spoken Language Processing, volume 1, pages 377-380, Denver, 2002.

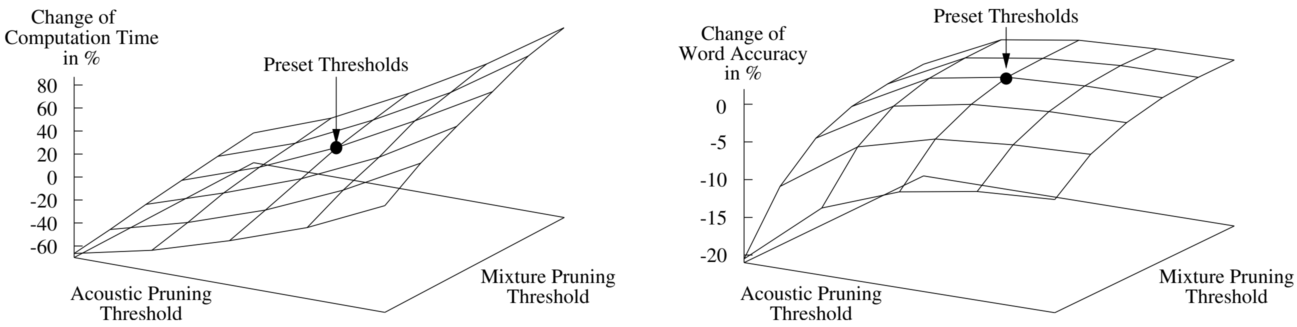

Fig. 1. Change of computation time and word accuracy with altered acoustic and mixture pruning thresholds, starting from preset values.

and started for the frames t_s to $t_s + \Delta t - 1$ are combined to a search tree group with t_s being an integral multiple of Δt. Within the group no competing state hypotheses are established, only the best hypothesis survives.

All the above mentioned techniques can be applied in one-pass as well as in multi-pass ASR systems. In multi-pass systems additionally the re-scoring of hypotheses by models of increasing complexity can be applied (cf. e.g. [7, 8]). However, in the following only time-synchronous ASR systems are considered as they are more suitable for real-time speech recognition.

3. PRUNING EXPERIMENTS

As explained in the previous section, beam search pruning is controlled by a single pruning threshold. By altering this threshold the complexity of the search space and thus the computation time is changed. By adjusting this threshold respectively the thresholds for several pruning techniques, it should be possible to process speech in a given amount of time. In order to determine the interdependencies between pruning parameter settings and both computation time and word accuracy, we carried out a series of experiments starting up from preset-thresholds we use normally. In this manner we analyzed mixture, acoustic, and language model pruning as well as merging of adjacent search trees on speech data from the German Verbmobil Corpus [9]. The idea behind these experiments is to create a search space, in which the optimal combination of pruning thresholds for a certain necessary or desired processing rate can be found.

In figure 1 the obtained computation times and word accuracies are shown for various combinations of acoustic and mixture pruning thresholds (all experiments were carried out on a Compaq Alpha EV6 with 500 MHz).

As could be expected lower thresholds lead to faster speech processing but also to decreased word accuracy. In contrast, higher thresholds lead to slower speech processing, while the word accuracy only slightly increases (thus the preset-thresholds are nearly optimal, if no time-constraints are given).

4. PRUNING CONTROL MECHANISM

Based on the results of the pruning experiments, which can be viewed as an estimation of computation time and word accuracy change for different pruning parameter combinations, we developed a mechanism to control the different pruning parameters dynamically.

First, we established a weighting parameter α to combine both threshold search spaces (for computation time and word accuracy) by calculating a score Q for the pruning parameter combinations:

$$Q = \alpha \cdot \Delta T \; - \; (1 - \alpha) \cdot \Delta WA \cdot \beta \qquad (2)$$

where ΔT is the measured change of computation time and ΔWA the measured change of word accuracy (both quoted in percentage, based on the preset-thresholds as described in the previous section) for a certain threshold combination. The value of α lies between 0 and 1. ΔWA is multiplied with a factor β (which was set to 20) to give the change of word accuracy a greater relevance, especially because for pruning parameter combinations near the preset-thresholds the change of computation time is much larger than the change of word accuracy.

The aim of this approach is to come up with a parameter to control the pruning thresholds, depending on whether fast computation or heigh word accuracy is desired. With this approach the search spaces created by the experiments in the previous section (see figure 1) can be combined into one search space, which shape depends on the value of the weighting parameter α (see figure 2): As higher the weighting parameter is chosen, as more the computation time determines the shape and vice versa. By minimizing the score in equation (2), that means by searching for the minimum within the combined threshold search space (as shown in figure 2), it is now possible to determine by the weighting parameter α whether pruning parameters with faster computation time or better word accuracy should be used.

Next we established a mechanism to determine, what value the weighting parameter α should take on. For this purpose, local measurements of processing time are performed in a certain interval. Such speed measurements are necessary to determine whether the system should be sped up or slowed down. Based on these time measurements and the following equations it is then possible to determine, how the weighting parameter α should be changed. If the measured processing time was longer than desired, the weighting parameter should be increased, so that pruning parameters with lower computation time are preferred and thus the system is sped up. Otherwise, if the system is faster than necessary, the weighting parameter α can be decreased. This leads to pruning parameter combinations with an increased computation time, but also with a possible higher word accuracy.

For this reason, the change of the weighting parameter α is calculated as follows:

- First, the current difference Δt_c between the required computation time t_r and the desired computation time t_d is calculated:

$$\Delta t_c = t_r - t_d \qquad (3)$$

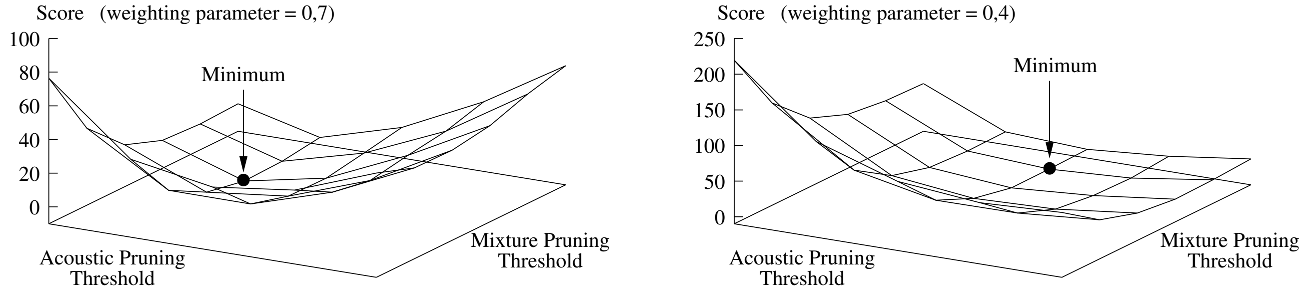

Fig. 2. Combined threshold search space for a weighting parameter α of 0.7 (left) and 0.4 (right).

- This time difference Δt_c determines the change of the weighting parameter α:

$$\Delta\alpha = \gamma \cdot (\Delta t_c - \Delta t_l) \tag{4}$$

  where γ is a constant and Δt_l is the last time difference, that is the time difference calculated at the previous local time measurement.

Equation (4) is only applied, if the current time difference is

1. positive and larger than the last time difference.
2. negative and smaller than the last time difference.

The reasons for this conditions are the following:

a) If the current computation time is greater (lower) than the desired computation time, the weighting parameter should be increased (decreased).

b) On the other hand, if the current time difference is positive but lower (negative but greater) than the time difference measured at the last time of regulation, the system has already been sped up (slowed down) and, therefore, there is no requirement to change the processing speed.

Additionally, by the constant γ in equation (4) it is possible to adjust the influence of the control mechanism. The smaller this constant is chosen, the slower the control mechanism reacts on time differences.

Finally, we extended the ASR system with the developed pruning control mechanism. In order to determine a time-constraint, a real-time factor (RTF, see equation (1)) has to be specified, so that the desired computation time for the speech input can be calculated. The necessary time differences are obtained by a local time measurement every tenth frame (frame length is 10 ms). To adjust the pruning-parameters, first equation (3) is computed. If the conditions for changing the weighting parameter α are met, equation (4) is computed. Afterwards, the score in equation (2) is minimized for the new weighting parameter (and therefore within the new combined threshold search space) by gradient descent, starting from the current threshold combination. The obtained threshold combination is then used for ongoing recognition.

5. EVALUATION

The ASR system extended by the control mechanism for dynamic search-space pruning (DSSP) was tested on two different recognition tasks.

Because the changes of computation time and word accuracy were measured on the German Verbmobil Corpus [9], the extended system was consequentially first evaluated on this recognition task. For this purpose, a 5336 word recognition system, trained on over 32 hours of spontaneous speech, was tested on the independent test-set with 343 utterances of approximately 41 minutes of spontaneous speech.

Additionally, we tested the system on the Wall-Street-Journal task (WSJ0) [10]. The control mechanism for dynamic search-space pruning likewise uses the changes of computation time and word accuracy measured on Verbmobil. The 5k closed vocabulary speaker independent recognition system used for this task was trained on about 15 hours of speech (the phonotypical transcription of the vocabulary was supplied by "Carnegie Mellon Pronouncing Dictionary" Version 0.6) and tested on 330 utterances with approximately 40 minutes of speech.

The feature extraction of the ASR system calculates every 10 ms 12 mel-frequency cepstral coefficients (MFCC) with forward masking [11], one energy term and the smoothed first and second order derivatives of these 13 features. On this 39 dimensional feature space semi-continuous HMM systems with linear structure, variable number of states and tri-phone sub-word units were trained. Additionally, bi-gram language models (perplexity: 64.4 on Verbmobil, 109.6 on WSJ0) were applied. For more details about the baseline ASR system ESMERALDA see [12].

Every experimental setting described in the following was tested nine times (on three different Compaq Alpha EV6 with 500 MHz, three times in each case) to obtain averaged computation times (expressed as RTF) and word accuracies. Additionally, on every utterance an individual RTF is measured, thus we can calculate the variance within the obtained over-all RTF.

Verbmobil			
System	over-all RTF	Variance of RTF	WA
without DSSP	2.0	0.17	75.91 %
with DSSP	2.0	0.02	75.71 %
	1.5	0.02	74.12 %
	1.0	0.01	69.95 %

Table 1. Word accuracies obtained by the ASR system with dynamic search-space pruning under different time constraints on Verbmobil. The first row shows the performance of the system with preset-thresholds and without DSSP.

Table 1 shows the results on Verbmobil for different settings. Although the over-all RTF without DSSP (and with preset-thresholds) is 2, only 13% of the utterances are effectively processed with an RTF of 2 (39% with an RTF between 1.9 and 2.1).

If the ASR system with DSSP and a given RTF of 2 is used, there is a small but not significant loss of WA which was expected, because the system with preset-thresholds and without DSSP delivers the nearly maximum WA (as mentioned in section 3). On the other hand, the variance within the over-all RTF is strongly reduced and over 62% of the utterances are processed within the given time (90% with an RTF between 1.9 and 2.1).

Furthermore, the ASR system with DSSP is able to process utterances in real-time (over-all RTF 1, only 6% slower than RTF 1.1). Thus the computation time can be halved, whereas the relative loss of word accuracy is only 7.9% on average.

In order to verify the results achieved on Verbmobil, we additionally carried out experiments on Wall-Street-Journal. For this experiments, the same estimated changes of computation time and word accuracy as for Verbmobil were used.

Wall-Street-Journal			
System	over-all RTF	Variance of RTF	WA
without DSSP	3.2	0.10	87.91 %
faster CPU	2.7	0.09	87.91 %
with DSSP	3.2	< 0.01	87.67 %
	2.0	< 0.01	85.31 %
	1.0	< 0.01	77.67 %
faster CPU	1.0	< 0.01	80.93 %

Table 2. Word accuracies obtained by the ASR system with dynamic search-space pruning under different time constraints on Wall-Street-Journal. The first and second row show the performance of the system with preset-thresholds and without DSSP.

The results in table 2 point out, that the ASR system with DSSP can also be applied on Wall-Street-Journal without separate time-consuming experiments to estimate the threshold search space on this Corpus. Because the loss of word accuracy is possibly to high under real-time processing on this Corpus, we made some evaluations on a faster CPU (Compaq Alpha EV6.7 with 667 MHz). The expected minor loss of word accuracy when using more processing power is automatically achieved applying the proposed DSSP technique.

6. CONCLUSION

In this paper we proposed a new technique for dynamic search-space pruning (DSSP) that makes time-constrained speech recognition possible. We developed a control mechanism to optimize different pruning parameters dynamically during the recognition process. With this control mechanism it is possible to process speech input with a given real-time factor. Furthermore, this control mechanism optimizes the pruning parameters with respect to the expected word accuracy.

For testing the ASR system extended with DSSP, the control mechanism adjusted acoustic and mixture pruning as well as merging of adjacent search trees. However, the proposed control mechanism can in principle include all kinds of pruning techniques, which are based on parameters that can be changed during the recognition process.

The extended ASR system has shown a reliable processing rate and an acceptable loss of word accuracy for different time-constraints. Compared to a system without DSSP, which has the same time available for computation, the variance of the processing rate is decreased greatly. Additionally, the DSSP technique can deliver real-time performance for non-trivial recognition tasks automatically if the baseline system achieves a reasonable performance.

7. ACKNOWLEDGEMENT

This research was supported by the German Research Foundation (DFG) within SFB 360 "Situated Artificial Communicators".

8. REFERENCES

[1] B.T. Lowerre, *The HARPY Speech Recognition System*, Ph.D. thesis, Carnegie-Mellon University, Department of Computer Science, Pittsburg, 1976.

[2] H. Ney and S. Ortmanns, "Dynamic programming search for continuous speech recognition," *IEEE Signal Processing Magazine*, vol. 16, no. 5, pp. 64–83, 1999.

[3] V. Steinbiss, B.-H. Tran, and H. Ney, "Improvements in beam search," in *Proc. Int. Conf. on Spoken Language Processing*, Yokohama, 1994, vol. 4, pp. 2143–2146.

[4] S. Ortmanns, A. Eiden, H. Ney, and N. Coenen, "Look-ahead techniques for fast beam search," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, München, 1997, vol. 3, pp. 1783–1786.

[5] J. Wu and F. Zheng, "Reducing time-synchronous beam search effort using stage based look-ahead and language model rank based pruning," in *Proc. Int. Conf. on Spoken Language Processing*, Bejing, 2000, vol. 4, pp. 262–265.

[6] G.A. Fink, C. Schillo, F. Kummert, and G. Sagerer, "Incremental speech recognition for multimodal interfaces," in *Proc. 24th Annual Conf. of the IEEE Industrial Electronics Society*, Aachen, 1998, pp. 2012–2017.

[7] V. Steinbiss, H. Ney, X. Aubert, S. Besling, C. Dugast, U. Essen, R. Haeb-Umbach, R. Kneser, H.-G. Meier, M. Oerder, and B.-H. Tran, "The Philips research system for continuous-speech recognition," *Philips Journal of Research*, vol. 49, no. 4, pp. 317–352, 1996.

[8] T. Hain, P.C. Woodland, T.R. Niesler, and E.W.D. Whittaker, "The 1998 HTK system for transcription of conversational telephone speech," in *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, Phoenix, Arizona, 1999, vol. 1, pp. 57–60.

[9] K. Kohler, G. Lex, M. Pätzold, M. Scheffers, A. Simpson, and W. Thon, "Handbuch zur Datenaufnahme und Transliteration in TP 14 von VERBMOBIL – 3.0," Tech. Rep. 11, Institut für Phonetik und digitale Sprachverarbeitung, Universität Kiel, 1994.

[10] D. B. Paul and J. M. Baker, "The design for the Wall Street Journal-based CSR corpus," in *Speech and Natural Language Workshop*. 1992, Morgan Kaufmann.

[11] S. Wendt, G.A. Fink, and F. Kummert, "Forward masking for increased robustness in automatic speech recognition," in *Proc. European Conf. on Speech Communication and Technology*, Aalborg, 2001, vol. 1, pp. 615–618.

[12] G.A. Fink, "Developing HMM-based recognizers with ESMERALDA," in *Lecture Notes in Artificial Intelligence*, V. Matoušek, P. Mautner, J. Ocelíková, and P. Sojka, Eds., Berlin Heidelberg, 1999, vol. 1692, pp. 229–234, Springer.