A ONE-PASS DECODER BASED ON POLYMORPHIC LINGUISTIC CONTEXT ASSIGNMENT

Hagen Soltau, Florian Metze, Christian Fügen, and Alex Waibel

Interactive Systems Laboratories
University of Karlsruhe (Germany), Carnegie Mellon University (USA)
{soltau|metze|fuegen|waibel}@ira.uka.de

ABSTRACT

In this study, we examine how fast decoding of conversational speech with large vocabularies profits from efficient use of linguistic information, i.e. language models and grammars. Based on a re-entrant single pronunciation prefix tree, we use the concept of linguistic context polymorphism to allow an early incorporation of language model information. This approach allows us to use all available language model information in a one-pass decoder, using the same engine to decode with statistical n-gram language models as well as context free grammars or re-scoring of lattices in an efficient way.
We compare this approach to our previous decoder, which needed three passes to incorporate all available information. The results on a very large vocabulary task show that the search can be speeded up by almost a factor of three, without introducing additional search errors.

1. INTRODUCTION

Recent work on search strategies for automatic speech recognition (ASR) has been directed towards single-pass decoding, even for large vocabulary tasks [8, 3, 5]. The reasoning behind this research is the potential advantage of applying all available knowledge sources as early as possible, which should make it possible to use tighter pruning thresholds, leading to a more precise beam search and therefore to more efficient decoding. Also, a one-pass decoding strategy is usually advantageous with respect to real-time requirements of many of today's ASR applications.

However, a careful organization of the search space is necessary in order to integrate cross-word acoustic models and long-span language models (LM) in one search pass. These methods contribute significantly to a recognizers performance and must therefore be retained when implementing a one-pass search strategy. Even then, early implementation of full language model information might be harmful (e.g. inaccurate models on unmatched domains) on high complexity tasks, where multi-pass search strategies with different language models can be employed.

In this paper, we describe the results of our comparison between a multi-pass search strategy and a one-pass search strategy on different tasks for the Janus ASR system. The results show that the proposed concept of polymorphic linguistic context assignment applied to a re-entrant pronunciation tree is particularly effective for decoding with very large vocabularies. Smaller systems improve, too, albeit not as much as larger systems, which are receiving a lot of attention in the context of unrestricted tasks such as meeting recognition [4].

The first section of this paper outlines different strategies employed for time-synchronous beam search in speech recognition. We describe in detail both multi-pass and single-pass decoding schemes. The next section covers a number of tasks and systems[1], that we tested our two decoders on. These experiments are described in the following and summarized in the last section.

2. DECODING STRATEGIES

The two decoding strategies in this paper base on a lexicon organized as a pronunciation prefix tree (PPT), where the search tree is traversed in a time synchronous way. A simple PPT is shown in figure 1. At each time frame, the active roots, nodes and leafs are expanded into their children and then pruned.

The main differences between the two approaches concern the access of linguistic information during the search and the recombination of different hypotheses to efficiently use linguistic constrains. These are given by the language model history in the case of statistical n-grams or state transitions for context-free grammars.

A PPT is an elegant and compact way to expand hypotheses within words. When reaching the end of a PPT (the leafs) it becomes necessary to extend the search to the following word candidates. This can be

[1] By this term, we mean a set of acoustic and language models

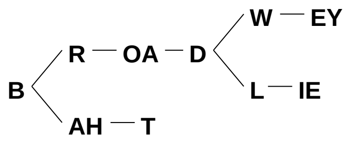

Figure 1: A simple pronunciation prefix tree (PPT) for a system using context-independent acoustic models.

done either by creating a copy of the tree or by re-entering the tree and keeping track of the word history.

2.1. Multi-Pass Decoding with delayed LM information

This approach uses a fast first search pass, which uses linguistic information only in an approximative way to constrain the search space sufficiently to allow for the application of more expensive search algorithms in subsequent passes.

The critical part in this approach, which is used in the standard Janus decoder, is the path recombination at the tree leafs. On entering a new tree root, only the best local predecessor word will be connected to the starting root node. Since the word identity is not yet known, the language model cannot be applied at this stage and the best predecessor will be determined without the language model information. To avoid unrecoverable search errors, wide beams have to be used and word segmentation will also be affected, since the correct starting point for a word depends on the predecessor word, which will be ignored here.

The language model is applied at the tree leafs, where the word identity is known. However, since only the local best predecessor is kept in memory, one can only correctly apply a bigram-LM. For higher-order language models, the back-pointer table has to be traced back in order to get the linguistic state ("poor-man's trigrams"). The second and third search pass are therefore used to correct the errors described above. To avoid additional acoustic score computations in the latter search passes, the acoustic scores from the first pass can be cached, which can require a significant amount of memory. In summary, the search strategy implemented in our old decoder is as follows:

1. search on tree-organized pronunciation lexicon
   - aggressive path recombination at word ends
   - use linguistic information only approximative
   - generate a list of starting words for each frame
2. search on flat-organized pronunciation lexicon
   - fix the word segmentation from the first pass
3. A-Star lattice re-scoring
   - full use of language model

2.2. One-Pass Decoder

To be able to include all available information sources in one pass, it is necessary to delay the inclusion of the full language model information until the word identity is known in the leafs of the tree. Then however, it is also possible to determine the best predecessor words, or the linguistic state, as we call it. One way to achieve this is the tree-copying process as described in [7, 8, 9, 3]. The idea is to create a separate copy of the PPT for each surviving linguistic state after path recombination. Since the number of different linguistic states (= tree copies) can be a few hundred for a long-span language model, efficient pruning criteria must be applied to fit computing and memory constraints. The search is therefore guided by a language model lookahead which distributes the language model probabilities over the PPT to allow a more efficient beam search.

In the following we describe an alternative approach to the tree copying process which allow a more efficient handling of the linguistic states. The underlying idea is to establish a linguistic polymorphism for each node of the PPT similar to the concept described in [1, 5]. The search space is then based on one single pronunciation prefix tree only:

- one copy of tree with dynamically allocated instances of nodes
- early path recombination
- full language model lookahead
- approach allows easy decoding along context-free grammars

In each node of the PPT, we keep a list of linguistic morphed instances. Each instance stores his own backpointer and scores for each state of the underlying Hidden Markov Model (HMM) with respect to the linguistic state of this instance. Since the linguistic state is known, we can apply the complete language model information for these scores, given the possible successor words for that node in the PPT. The LM scores will be updated on demand based on the compressed PPT. An example for the linguistic morphed instances within the PPT framework is shown in figure 2.

The advantage of this search space organization is that we can apply a beam and topN pruning strategy for the list of instances in a very easy way. This allows us to overcome the subtree dominance problem for the tree-copying approach. If there are two instances of a node where the linguistic state of one instance cause a worse LM score for the best possible successor word

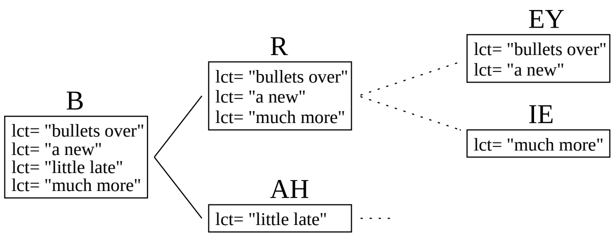

Figure 2: linguistic morphed instances within the PPT framework. (LCT= linguistic context)

compared to the LM score for the worst possible successor given the linguistic state of the other instance, the instance can already be eliminated.
Additionally, we perform the path recombination (which is usually done at the word ends) as soon as the word becomes unique, which is usually a few nodes before reaching the leaf. This is in particular useful in combination with the use of cross-word models. It is important to keep the number of instances in the leafs as small as possible to reduce the computational effort due the fan-out of the right context models.
This search space organization offers also advantages in terms of memory usage. Only a very tiny tree skeleton will be created permanently. The main memory require the instances which will be allocated dynamically on demand. Since the number of instances per node decrease very rapidally with the tree level, the search space can be handled very flexible and scalable.
To run the decoder with arbitrary linguistic knowledge sources as statistical n-grams, context free grammars, or word graphs, we use an abstract interface between the decoder and the linguistic knowledge sources. The interface consists of few functions to manipulate the linguistic state. The decoder itself works independent from the actual linguistic knowledge source.

2.3. Cross-Word Modeling

To use cross-word models for the tree roots, we apply the same concept that we used to handle different linguistic contexts here now at the HMM state level for different left phonetic contexts. However, we do not keep really a list of different state instances but only the local best left phonetic context. The integration of cross-word models at the tree leafs needs more computational effort, since we have to compute the scores for each possible right phonetic context. The number of different right phonetic context instances can be reduced by using a dual map between the phonetic context and the unique acoustic models. Depending on the phones set and the context decision tree, the fan-out can be reduced by factor of more than two on average. Another reduction of the fan out can be achieved by using the early path recombination as described above.

3. EXPERIMENTAL SETUP

The experiments described in this work were conducted on three tasks, using two different ASR systems for both English and German.

The first task is is the final test-set of the German Verbmobil-II project ("GSST"); it features conversational speech on a limited domain under relatively clean acoustic conditions. The second task is read speech from the Broadcast News (BN) corpus; it consists of clean, read speech from a very large domain. The third task is a subset of the Meeting data set [4], which was recorded at informal group meetings, containing very colloquial speech recorded through lapel microphones.

Database	VM-II	BN	Meeting
Speaking style	convers.	read	colloquial
Train speech data	62h	100h	
LM corpus	670k	141m	
Vocabulary	10k	40k	
Test speech data	65min	20min	55min

Table 1: Systems used in the comparison experiment.

The acoustic and language models we used to decode this data were taken from a preliminary ISL system for the final 2000 VM evaluation [11] without semitied covariances and feature space adaptation in the case of the German data and from a system trained on Broadcast News and English Verbmobil-II ("ESST") acoustics and a collection of text sources[2] in the case of the "Read BN" and "Meeting" tasks. For the German system, we use 3300 context dependent acoustic models with 167k gaussians and 4000 models with 132k gaussians for the English system. A summary of the system characteristics is shown in table 1 and 2.

System/ PPT	# roots	# nodes	# leafs
GSST	679	36651	11355
BN/Meeting	1159	103114	45095

Table 2: search tree size (# = number of)

4. EXPERIMENTS

Our results are shown in table 3. All timings were obtained on a standard PC with a Intel PIII/600 processor. No speed-ups such as the Bucket Box Intersection Algorithm[6] or phonetic fast match were used. The slightly improved error rates for the English system (0.8% for read BN and 0.3% for the Meeting task)

[2] BN, ESST, Crossfire, Newshour, WSJ.

are a result of a different handling of single phone words which are more important for English than for German.

Database	VM-II		BN		Meeting	
Decoder	3-p	1-p	3-p	1-p	3-p	1-p
RTF	6.8	4.0	12.2	4.2	55	38
WER (%)	26.9	26.9	14.7	13.9	43.7	43.4

Table 3: Comparison experiments between the two Janus decoders. (RTF = real time factor, 3-p = three pass decoder, 1-p = one pass, single prefix tree decoder)

Besides the real time factors, the number of active instances is also interesting to get a impression about the active search space. One can see, that the average number of instances is very moderate. Even for a very complex meeting task with unmatched acoustic and language models, the number of instances drops very rapidally. The small number of instances allows us to avoid any approximations for the language model range for the lookahead. In all experiments, we used the complete trigram history for the lookahead tree.

Task	# roots	# nodes	# leafs
GSST	231 (3.5)	253 (2.2)	19 (1.7)
read BN	274 (2.6)	298 (1.7)	17 (1.4)
Meeting	845 (8.6)	5037 (2.9)	219 (1.9)

Table 4: active search space (# = average number of models (instances) per 10ms)

5. SUMMARY

In this work, we have compared two different decoding strategies in the same environment on different tasks. Our results show that the early integration of full language model information is indeed helpful with respect to overall decoding effort on tasks with relatively low complexity. The greatest speed-up, close to a factor of three, was achieved on the "Read BN" task with a 40k vocabulary and clean acoustics. On the "GSST" task with a 10k vocabulary and spontaneous speech, the one-pass decoder runs in 60% of the time of the multi-pass decoder. On difficult tasks such as "Meeting", the one-pass search strategy still allows for savings in time and memory. However, the speed-up depends strongly on matched domain conditions.

As the importance of language models increases with their early integration in the search process, future work will be directed towards on-line language model and vocabulary adaptation as well as the combination of language models and grammars for efficient decoding.

6. REFERENCES

[1] F. Alleva, X. Huang, and M.-Y. Hwang. Improvements of the pronunciation prefix tree search organization. In *Proceedings of the ICASSP*, Atlanta, USA, 1996.

[2] G. Antoniol, F. Brugnara, M. Cettolo, and M. Federico. Language model representations for beam search decoding. In *Proceedings of the ICASSP*, Detroit, USA, 1995.

[3] X. Aubert. One pass cross word decoding for large vocabularies based on a lexical tree search organization. In *Proceedings of the Eurospeech*, Budapest, Hungary, 1999.

[4] Alex Waibel et.al. Advances in meeting recogniton. In *Proceedings of the First International Conference on Human Language Technology Conference (HLT 2001)*, San Diego, USA, 2001.

[5] M. Finke, D. Koll, J. Fritsch, and A. Waibel. Modeling and efficient decoding of large vocabulary conversational speech. In *Proceedings of the Eurospeech*, Budapest, Hungary, 1999.

[6] J. Fritsch and I. Rogina. The bucket box intersection (BBI) algorithm for fast approximative evaluation of diagonal mixture gaussians. In *Proceedings of the ICASSP*, Atlanta, USA, 1996.

[7] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder. Improvements in beam search for 10000-word continuous speech recognition. In *Proceedings of the ICASSP*, San Francisco, USA, 1992.

[8] J. Odell. *The Use of Context in Large Vocabulary Speech Recognition.* PhD thesis, University of Cambridge, United Kingdom, 1995.

[9] S. Ortmanns, A. Eiden, and H. Ney. Improved lexical tree search for large vocabulary speech recognition. In *Proceedings of the ICASSP*, Seattle, USA, 1998.

[10] M. Ravishankar. *Efficient Algorithms for Speech Recognition.* PhD thesis, Carnegie Mellon University, USA, 1996.

[11] H. Soltau, T. Schaaf, F.Metze, and A. Waibel. The ISL evaluation system for Verbmobil-II. In *Proceedings of the ICASSP*, Salt Lkae City, USA, 2001.

[12] M. Woszcyna and M. Finke. Minimizing search errors due to delayed bigrams in real-time speech recognition systems. In *Proceedings of the ICASSP*, Atlanta, USA, 1996.