A COMPARATIVE ANALYSIS OF DYNAMIC NETWORK DECODING

David Rybach, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition, Computer Science Department, RWTH Aachen University, 52056 Aachen, Germany

{rybach, schlueter, ney}@cs.rwth-aachen.de

ABSTRACT

The use of statically compiled search networks for ASR systems using huge vocabularies and complex language models often becomes challenging in terms of memory requirements. Dynamic network decoders introduce additional computations in favor of significantly lower memory consumption. In this paper we investigate the properties of two well-known search strategies for dynamic network decoding, namely history conditioned tree search and WFST-based search using dynamic transducer composition. We analyze the impact of the differences in search graph representation, search space structure, and language model look-ahead techniques. Experiments on an LVCSR task illustrate the influence of the compared properties.

Index Terms- LVCSR, WFST, HCLT, beam search

1. INTRODUCTION

Recent LVCSR systems for unconstrained domains, morphologically rich languages, and spontaneous speech input use huge vocabularies and complex language models. Decoders using static search networks, which are known to be very efficient, may exceed memory constraints, especially if the language model (LM) used for decoding is not simplified. Dynamic network decoders, which generate the required parts of the search graph on demand during the search, have much lower memory requirements.

In this paper, we compare two search strategies for dynamic network decoding. The history conditioned lexical tree (HCLT) search is a well known strategy for dynamic network expansion [1]. Weighted finite state transducers (WFST) are widely used for the construction and representation of static search networks [2]. Lazy (or on-the-fly) transducer composition [3] allows us to use the WFST framework also for dynamic network decoding. We use the concept of composition filters [4] for efficient dynamic composition. Other methods for WFST-based dynamic network decoding, like on-the-fly hypothesis rescoring [5] are not part of our analysis.

In [6] the HCLT search is compared to decoding using a static WFST-based search network. The authors of [7] describe optimization techniques for a dynamic network decoder and compare its performance to a static WFST-based decoder. In contrast to publications comparing static to dynamic network decoding, this paper focuses on the particular properties of the search strategies for dynamic network decoding.

We used our well-proven HCLT decoder [8] and a recently developed WFST-based decoder, which uses the Open-Fst toolkit [9], to investigate the impact of the search strategies' distinctive properties on an LVCSR task.

Section 2 gives a review of the investigated search strategies. In Section 3 we describe their differences. Section 4 presents the experimental results which are concluded in Section 5.

2. SEARCH STRATEGIES

In the search process, the most likely word sequence for the observed speech signal is determined by applying a one-pass search strategy based on dynamic programming [10]. In a dynamic network decoder, the search space is constructed dynamically by integrating parts of the LM as needed during the search. Beam search avoids an exhaustive search and focusses the search on the most likely partial hypotheses.

2.1. HCLT Decoding

The history conditioned lexical tree decoder uses a lexical prefix tree as representation of the pronunciation dictionary [1, 10]. In our decoder, the so called *state tree* is constructed using the tied HMM state sequences of the pronunciations for the words in the vocabulary. Words occur at the leaf nodes of the tree.

In order to apply the probabilities of the LM, search hypotheses have to be separated by their predecessor words. For an *n*-gram LM, only the most recent n-1 words have to be considered. Thus, the search space is structured by history conditioned "tree copies". A physical copy of the tree is obviously not required, but rather the computation of a table Q(t, h, s) where t is the time frame, h a history, and s is a state in the prefix tree. Q(t, h, s) is the score of the best hypothesis at time t ending in state s of the tree copy for history h. At word boundaries the recombination is carried out on the word level, merging hypotheses with equivalent histories.

Pruning is performed both on the state level and the word end level. The pruning of word ends limits the number of active tree copies. The acoustic pruning can be refined by incorporating the LM probabilities as early as possible using *language model look-ahead* [11]. The anticipated LM probability for a certain state in the tree is approximated by the best word end reachable. For efficiency only a truncated history is used to compute the LM scores. This look-ahead score is incorporated in the pruning process by combining it with the probability of the state hypothesis.

This work has been funded in part by the Google Research Awards Program and was partly realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

For the use of across word context dependent models the tree structure has to be modified [12]. At word ends, different states are introduced for all possible successor phones, forming a fan-out. The root state is split into states for every phone pair. From the fan-out states only valid transitions to one of the root states are considered.

2.2. WFST-based Decoding

In the WFST-framework all knowledge sources are represented by weighted finite state transducers. The LM is represented by a transducer G, L is a phone to word transducer derived from the pronunciation dictionary, and C encodes the context dependency of the acoustic models. These transducers are combined by the finite-state operation of composition as $C \circ L \circ G$. This composed transducer has tied HMM labels on the input side and words as output labels. The HMM states are generated dynamically during decoding in our system.

The composition of two transducers can be performed using lazy evaluation, by computing the arcs of a state only when it is discovered [3]. In the dynamic network decoder, the composition of $(C \circ L)$ with G is computed on-the-fly. Several issues have to be considered for this composition to be efficient, which are solved by using special composition filters [4, 13]. These composition filters provide an online pushing of labels and weights, which prevents the expansion of useless paths and provides an early incorporation of LM scores using weight pushing.

Using the same notation as in the previous section, the decoder computes $Q(t, (s_L, s_G))$ where (s_L, s_G) is a tuple of states from the $(C \circ L)$ and the G transducer respectively. The decoder itself does not need to distinguish between a static transducer and a dynamically composed one. The LM recombination and the weight pushing are performed transparently by the on-the-fly composition procedure.

3. COMPARISON

The search strategies described in the previous section share the same principles, but differ in several aspects described in the following sections.

3.1. Search Graph

The $C \circ L$ transducer and the state tree both encode the pronunciation dictionary with context dependent phone models. A determinized L is equivalent to a lexical prefix tree on context independent phone level. Context dependency – within words and across words – is introduced by the composition with C after building the closure of L. An additional minimization of L merges common suffixes.

The state tree is determinized on the tied HMM state level, common suffixes are not shared. The fan-out and fan-in required for across word modelling do not account for the HMM state model tying, thereby increasing the size of the search graph.

3.2. Search Space Structure

In the HCLT search, the search space is structured by word histories of length n -1 for an n-gram LM. All words of the vocabulary can be reached from the root state of the tree, thus the full history is required to compute their LM probabilities.

The sparsity of the LM is not exploited in this strategy. The WFST-based search structures the search space by states in the G transducer which accounts for the backing-off in the LM. However, a word can be hypothesized in both the full context and in all backing-off states reachable by epsilon transitions. Due to the label pushing and minimization, hypotheses can be recombined before the actual word end is reached.

The potential search space of the HCLT strategy has a size of $O(V^{n-1} \cdot S)$ with V the size of the vocabulary, n the order of the LM, and S the size of the state tree. The size of the search space for the WFST strategy depends on the size of the G transducer which is proportional to the number of n-grams in the LM. However, only a small fraction of the search space needs to be generated during the decoding.

3.3. Active Search Space

The size of the active search space depends on the degree of redundancy in the search graph and – more importantly – on an effective pruning. The early incorporation of the LM is crucial for distinguishing promising hypotheses from unlikely ones. The HCLT decoder's LM look-ahead uses a truncated history to calculate an approximative LM score. In contrast, the WFST-based decoder can use the full history. Furthermore, the look-ahead considers only the score of the best reachable word, whereas weight pushing accounts for the score of all reachable words (if the log semiring is used).

Experiments showed that it is important to incorporate the look-ahead score as early as possible. Our decoders prune state hypotheses already before the calculation of acoustic likelihoods using anticipated pruning taking the look-ahead or arc weight into account.

3.4. Word Boundaries

Word boundaries are obvious in the HCLT search space, which facilitates the assignment of word start and end times as well as the generation of word lattices. In the transducer, word labels can be moved before or behind the last phone of a word by transducer optimizations or the composition with a shifted C transducer. Leaving an arc with an output label does not necessarily correspond to the true word end. In our WFST-based decoder, we exploit the word boundary information of the context dependent phone models to detect transitions between words.

3.5. Non-speech Events

Non-speech events, like silence and noise, cause problems, because they do not perfectly fit in the transducer-based framework, if they are not part of the LM [14]. If noise and silence tokens are required in the decoder output, they have to occur in the G transducer. Furthermore, if weight pushing using the log semiring is used, then we have to assign them some weight [15]. If non-speech events shall preserve the language model context, loop transitions have to be added for every state in G and each non-speech event, increasing the size of G. The HCLT decoder however does not require LM scores for all tokens.

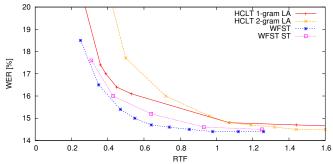


Fig. 1. Real-time factors for the HCLT and the WFST decoder using different search graphs and look-ahead (LA) sizes.

4. EXPERIMENTAL RESULTS

We evaluated the influence of the discussed differences using our LVCSR system for English parliamentary speeches. The ASR system is described in the following section, followed by a presentation and discussion of the experiments.

4.1. Recognition System

The decoders use a baseline acoustic model (AM) for the automatic transcription of European Parliament Plenary Sessions (EPPS) in English as described in [16]. The dictionary contains 53K words with 59K pronunciations, modeled using 45 phones and 6 non-speech pseudo-phones. The AM consists of 900K densities for 4500 Gaussian mixtures modelling generalized triphone states with across word context dependency and using word boundary information. The 4-gram LM used contains 7.4M n-grams. The test set comprises 644 segments with a total duration of 2.85h with about 27K words in total. The runtime measurements were performed on a 2.8 GHz Intel Core2.

4.2. Experiments

In a first experiment we compared the runtime efficiency of both decoders. Figure 1 shows the word error rate (WER) as function of the real time factor (RTF), i.e. processing time divided by audio duration. The results show that the WFSTbased decoder runs about 40% faster achieving the same WER. A detailed profiling (see Figure 2) unveils that in both decoders ~60% of the computation time is spent for acoustic score computations (quantization, SIMD instructions, and batched computations are used). The runtime required for the composition (20%) is less than the sum of the costs for word end-handling (11%), i.e. LM score computation and recombination, and LM look-ahead score computation (19%). The state expansion in the WFST-based decoder requires more computation time (15% vs. 6%) because of the dynamic expansion of HMM states.

Figure 3 shows that the number of state hypotheses is very similar for both decoders. The number of (batched) acoustic score computations in the HCLT decoder is 25% higher though, due to more variation in the hypothesized HMM state models, which causes the difference in absolute AM computation time in Figure 2.

In order to analyze the effect of differences in the search graph size, we converted the state tree of the HCLT decoder to a transducer and used it instead of $C \circ L$ (denoted by WFST

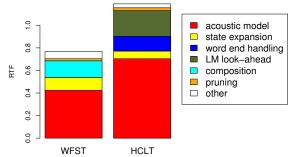


Fig. 2. Profiling of the HCLT and the WFST decoder. The respective pruning values were chosen to yield the same WER.

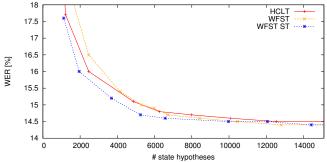


Fig. 3. WER vs. average number of active state hypotheses.

ST). The conversion required adding closure transitions for all word ends in the state tree. We applied factorization to exploit the decoder's dynamic expansion of HMM states. The size of these graphs is shown in Table 1. The state tree based graphs are larger, because of the fan-in and fan-outs for context dependent word transitions. The higher number of arcs in the state tree transducer compared to the regular state tree is due to the closure transitions for each word end in the tree. From Figure 3 can be seen, that the number of state hypotheses is about the same or even slightly lower compared to the smaller $C \circ L$ transducer. However, the number of active arcs is higher in the state tree transducer and the higher out-degree requires more computations, resulting in a higher RTF (cf. Figure 1).

The active part of the search graph is only a small fraction of the potential search graph. The HCLT search has an average number of 60 active tree copies. From the 37.6M states in the composed transducer of the WFST search, only 624K states are visited on average during the decoding of a segment.

The impact of the LM look-ahead and weight-pushing is shown in Figure 4. Here we included an experiment with fullhistory LM look-ahead scores, which results in a too high RTF for practical use, but illustrates the impact of the truncated histories. Furthermore, the graph shows the impact of the semiring used for weight pushing in the WFST transducer. Both full histories and the log semiring result in a lower number of state hypotheses. The number of state hypotheses in the HCLT search using the full history is lower compared to the WFST decoder, which can be explained by the additional word-end pruning step. Using a unigram look-ahead increases the active search space significantly, but the computation of the look-ahead scores is faster.

Table 1. Search graph size and active search space for HCLT search, WFST search, and WFST using the state tree (ST). The search graph size is given for the expanded (HMM state level) transducers (without G). The number of active state hypotheses is given for experiments yielding the same WER and using the same pruning threshold f respectively.

and using the same praiming theshold j respectively.				
			state hypotheses	
	states	arcs	WER=14.7	f = 250
HCLT	2,796K	2,857K	7,995	4,863
WFST ST	1,350K	4,122K	5,247	6,571
WFST	440K	588K	6,715	8,423

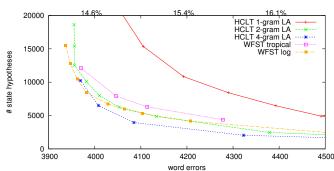


Fig. 4. Number of active states hypotheses as function of the absolute number of word errors for different LM look-ahead (LA) histories and semirings used for weight pushing. The corresponding WER is shown at the top of the plot.

5. CONCLUSIONS

The experiments in the previous section show that the search graph, the search space structure, and the early incorporation of LM probabilities affect the decoder performance, although only slightly. The search graph of the WFST decoder benefits from the smaller fan-out for across word transitions, which do not enlarge the number of state hypotheses but increase the costs for state expansions. We showed that the usage of the full history as well as the incorporation of all reachable words improve the LM look-ahead.

Moreover, the conceptually simpler structure of the WFST decoder facilitates a more optimized implementation. In the HCLT decoder, the complexity is shifted from the network construction to the decoder itself. Especially the computation of look-ahead tables and the handling of word ends require more computation time than the dynamic composition using look-ahead composition filters. Nevertheless, not all of these problems are inherent. The LM look-ahead can be performed more efficiently, as described in [7] and [17]. The authors of [7] also describe methods for minimized across-word transitions and early path recombination, which are currently not used in our system.

Further investigations should analyze larger vocabularies, more complex LMs, and the implications of the search strategies for lattice construction and multi-pass decoding.

6. REFERENCES

 H. Ney, R. Haeb-Umbach, B. Tran, and M. Oerder, "Improvements in beam search for 10000-word continuous speech recognition," in *ICASSP*, San Francisco, CA, USA, Mar. 1992, pp. 9–12.

- [2] M. Mohri, F. Pereira, and M. Riley, "Speech recognition with weighted finite-state transducers," in *Handbook of Speech Processing*, J. Benesty, M. Sondhi, and Y. Huang, Eds. Springer, 2008, ch. 28, pp. 559–582.
- [3] M. Mohri, F. Pereira, and M.Riley, "Weighted automata in text and speech processing," in *European Conference on Artificial Intelligence, Workshop on Extended Finite State Models* of Language, Budapest, Hungary, Aug. 1996.
- [4] C. Allauzen, M. Riley, and J. Schalkwyk, "A generalized composition algorithm for weighted finite-state transducers," in *INTERSPEECH*, Brighton, U.K., Sep. 2009, pp. 1203–1206.
- [5] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 4, pp. 1352–1365, 2007.
- [6] S. Kanthak, H. Ney, M. Riley, and M. Mohri, "A comparison of two LVR search optimization techniques," in *ICSLP*, Denver, CO, USA, Sep. 2002, pp. 1309–1312.
- [7] H. Soltau and G. Saon, "Dynamic network decoding revisited," in ASRU, Merano, Italy, Dec. 2009, pp. 276–281.
- [8] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, and H. Ney, "The RWTH Aachen University open source speech recognition system," in *INTERSPEECH*, Brighton, U.K., Sep. 2009, pp. 2111–2114.
- [9] C. Allauzen, M. Riley, J. Schalkwyk, W. Skut, and M. Mohri, "OpenFst: a general and efficient weighted finite-state transducer library," in *CIAA*, Prague, Czech Republic, Jul. 2007, pp. 11–23.
- [10] H. Ney and S. Ortmanns, "Progress in dynamic programming search for LVCSR," *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1224–1240, Aug. 2000.
- [11] S. Ortmanns and H. Ney, "Look-ahead techniques for fast beam search," *Computer Speech and Language*, vol. 14, no. 1, pp. 15–32, Jan. 2000.
- [12] A. Sixtus and H. Ney, "From within-word model search to across-word model search in large vocabulary continuous speech recognition," *Computer Speech and Language*, vol. 16, no. 2, pp. 245–271, May 2002.
- [13] C. Allauzen, M. Riley, and J. Schalkwyk, "Filters for efficient composition of weighted finite-state transducers," in *CIAA*, Winnipeg, Canada, Aug. 2010.
- [14] P. Garner, "Silence models in weighted finite-state transducers," in *INTERSPEECH*, Brisbane, Australia, Sep. 2008, pp. 1817–1820.
- [15] C. Allauzen, M. Mohr, B. Roark, and M. Riley, "A generalized construction of integrated speech recognition transducers," in *INTERSPEECH*, Montreal, Canada, May 2004, pp. 761–764.
- [16] J. Lööf, C. Gollan, S. Hahn, G. Heigold, B. Hoffmeister, C. Plahl, D. Rybach, R. Schlüter, and H. Ney, "The RWTH 2007 TC-STAR evaluation system for European English and Spanish," in *INTERSPEECH*, Antwerp, Belgium, Aug. 2007, pp. 2145–2148.
- [17] D. Nolden, H. Ney, and R. Schüter, "Exploiting sparseness of backing-off language models for efficient look-ahead in LVCSR," in *ICASSP*, Prague, Czech Republic, May 2011.