A *WAVE DECODER* FOR CONTINUOUS SPEECH RECOGNITION

Eric Burhke, Wu Chou and Qiru Zhou

Bell Laboratories
Lucent Technologies
600 Mountain Avenue
Murray Hill, New Jersey, NJ 07974, U.S.A.
Email: wuchou@research.att.com

ABSTRACT

In this paper, a *wave decoder* based on the general re-entrant network for continuous speech recognition is described. The decoder design is based on the concept of self-adjusting decoding graph in which the decoding network is expanded and released frame-synchronously. The fast network expansion and release are made possible by utilizing a novel dynamic network scaffolding layer. The self-adjusting decoding graph is obtained by slicing the traditional decoding network horizontally for separation of different knowledge sources and vertically according to each time instant in search. A two layer hashing structure and an admissible arc predication scheme are described. These methods significantly reduce the arc mortality rate, a problem which plagues the efficiency of the dynamic decoder. Experimental results demonstrate that an order of magnitude reduction of decoding resources can be achieved based on the proposed approach.

1. INTRODUCTION

In general, the search methods in speech recognition can be categorized into two classes, namely the best first search such as the stack decoder[7] and the breadth first search such as the traditional frame synchronous decoder[1]. Although it is relatively easy to incorporate long span language models in the best first search, the stack decoder in general is frame-asynchronous and requires good search heuristics to estimate the least upper bound of the path score. These search heuristics are important in order to maintain good search accuracy as well as for reducing the search complexity. On the other hand, in the breadth first search, search heuristics are not required and the search can be made frame synchronous, an important feature for many real-time applications. However, one of the drawbacks of the breadth first search is the required resources in order to support the search process. The use of cross-word tri-phone models and long span language models often lead to an order of magnitude increase in the size of decoding network if a static pre-compiled decoding network is used to support the search.

Many schemes have been proposed to reduce the decoding resources in order to support the use of various knowledge sources in decoding [4][5]. In particular, a one-pass dynamic decoder design was proposed in [3] which had many successes in making it possible to integrate long span language models and detailed acoustic models in various tasks with manageable decoding resource requirements. This approach employs a dynamic decoding network expansion scheme and is based on the assumption that the decoding network is non re-entrant. A non re-entrant decoding network is a network without looping back re-entrant arcs. However, in most applications, the decoding network is often compactly represented by a general re-entrant decoding networks. Mapping a re-entrant decoding network to a non re-entrant decoding work will lead to a network expansion and in theory, the size of the decoding network under this transformation can be unbounded. In many real applications, it is typical that the size of the vocabulary is not large but the grammar constraints, such as fixed length, parity check and etc., are very complicated. As a consequence, such a mapping is not only difficult but also expansive because of the increased size of the transformed network. In some cases, it may not produce the desired memory savings for medium or small vocabulary tasks, and it is difficult to incorporate complicated grammar constraints. Another concern is the rapid network growth in noisy environment. Under the mismatched conditions and spontaneous speech with noisy background, the size of the dynamically grown non re-entrant decoding network can become quite unstable, and careful control of the decoding network growth in adverse conditions becomes critical.

In this paper, a "wave decoder" based on a general re-entrant decoding network is proposed and the constraint of using a non re-entrant network in single pass dynamic decoding is eliminated. It offers more deeper pruning than the previous approach by using a more detailed integration of acoustic model matching with dynamic decoding network expansion.

2. A LAYERED DECODING NETWORK ARCHITECTURE

One observation made on the decoding network in speech recognition is that it is very clumsy, because various knowledge sources are mapped into its structure. This not only makes it difficult to expand the network dynamically but

also makes it expensive to locally rebuild the needed pieces in order to sustain the search. The proposed *wave decoder* is based on a dynamically constructed scaffolding (skeleton) layer which serves as the anchor point layer to support fast network expansion and reconstruction. Fig 1. illustrates the decoding network structure used in *wave decoder*. The decoding network in our approach is sliced horizontally into three layers: word net layer, phone net layer and the DP (dynamic programming) layer. The word net layer keeps the word level knowledge sources such as grammar constraints and specifications from the language model. Phone net layer keeps the phonetic level knowledge sources such as phonetic lexicon graph, etc. The DP net layer is based on the integration of all knowledge sources and is the place where the dynamic programming is performed.

The word net and phone net are independent from the acoustic model and do not carry any time information. The use of the cross-word tri-phone models and other detailed acoustic modeling techniques in search will only increase the size and the complexity of the DP net. But it has no effect on the upper two layers. Since the penalty on the phone transitions can be mapped down to the DP net, the phone net in fact can be made independent of both acoustic model and language model, a structure which can be easily copied and shared. Such a horizontal slicing of the decoding network has a very nice property that more than 90% memory usage is at the lower DP layer. The upper two layers form a very light scaffolding structure upon which very fast dynamic DP net expansion and releasing can be supported. In *wave decoder*, the decoding network is both constructed dynamically and released frame synchronously. Dynamically grown the scaffolding structure in the *wave decoder* is relatively easy, because the horizontal slicing of the decoding network in *wave decoder* not only makes the scaffolding structure light but also isolates the knowledge sources making it easy to grow. The real issue is the dynamic expanding and releasing of the DP net for a re-entrant decoding network, which is the topic of the next section.

3. DYNAMIC NETWORK PROPAGATING IN RE-ENTRANT NETWORK

The basic idea used in *wave decoder* is to make the decoding network (in particular the DP net) into a self-adjusting graph which, at any time instant, the memory space resources are self-adjusted to serve the need of that particular time frame. If viewed from the time axis, this corresponds to a vertical slice of the decoding network, $Net\_active(t)$, as illustrated by the unmarked area of DP net in Fig 1. $Net\_active(t)$ is the active portion of the decoding network at that particular time instant that the self-adjusting dynamic search graph represents. It is determined mainly by the model matching at that particular time frame and is relatively independent of the depth of the grammar and the duration of the speech input. In wave decoder,

$$\max(size(Net\_wave)) \approx \max(size(Net\_active(t))) \leq Net\_static. \quad (1)$$

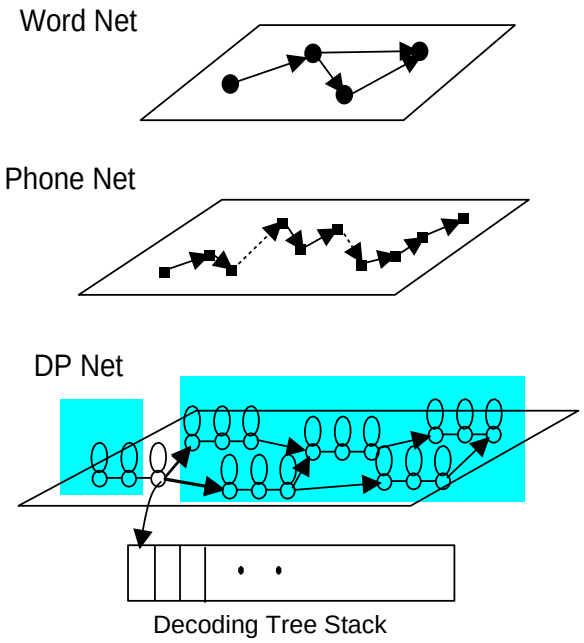

Figure 1: A layered decoding network architecture.

This is because the re-entrant to non re-entrant network transformation can be eliminated in the *wave decoder*.

The DP net forms the search graph in our approach and each active path is represented as a node in the corresponding decoding (search) tree of that search graph[8]. Each node in the decoding tree carries with itself all necessary links, scores and trace-back informations. Such a search graph structure has been used in static decoder for many years. However, in a dynamic decoder, the search graph is not physically built but the subset of the search graph covering the need of the decoding tree at the current time instant must be well established in order to make the search admissible. This dynamic net is self-adjusting and varies with time. When using a non re-entrant network, the expansion is performed forward and locally the network is not punctured by dynamic network releasing. In a re-entrant decoding network, it is often that the used network is reentered which is partially or totally released from the previous actions. One phenomenon is the so called "wanton expansion" where the network generation engine can not locate the remaining pieces of the used network and generate duplicated structures. This can incur significant overhead in memory usage and CPU time.

In order to use the re-entrant network in a single pass dynamic decoder, a two level dynamic hashing structure is used at the phone node of the scaffolding layer. It hashes the existing node of the DP net related to this phone node in its node table and it hashes the existing arcs of the DP net related to this phone node in its arc table. Beam search is performed for all nodes in the decoding tree. Network prun-

ing procedure is then followed to self-adjusting the shape of the dynamic search graph. It provides a minimum and sufficient decoding graph covering the needs of supporting the remaining active nodes in the pruned decoding tree.

DP Network Pruning

There are three structures in the *wave decoder* that are managed dynamically for the DP net. These are dynamic programming arcs, dynamic programming nodes and the hashing tables. The dynamic programming arcs are released once they are not an active part within the search beam. The dynamic programming nodes are released when they no longer terminate an active dynamic programming arc. The dynamic hashing tables, which are used to hold the record of existing portion of arcs and nodes under the scaffolding layer, are released when they become empty.

When reentering a used decoding network, the dynamic network generating engine first examines the node hash table at the scaffolding phone node to re-establish the missing nodes and reuse the existing nodes which have been left from the previous time instant. Then it examines the arc hashing table at the scaffolding phone node to re-establish the missing arc connections and reuse the existing arcs. The use of fast dynamic hashing is important because linear search can incur quite a cost in recovering a punctured network.

Arc Prediction

One issue closely related to the efficiency of the "wave decoder" is the mortality rate of newly created arcs and nodes in the DP net. In general, this issue can be more easily handled using fast-match and other heuristics in the network pruning procedure, although the admissibility of the search may suffer. In *wave decoder*, an admissible arc predication scheme is incorporated if no other heuristics for arc prediction are available. This scheme is based on a novel modification of the traditional beam search. In the traditional beam search, the best path score at the current time frame is first identified and paths, whose scores are more than a distance Δ below the best path, are pruned. However, computing the best path score in a dynamic decoder is not trivial. If a node is active, all scores of paths entering the first state of the following arcs must be calculated in order to determine the beam. However, many of the following arcs are often missing requiring to be rebuilt first. On the other hand, it is often that many of these newly built arcs are found out of the beam right after the best path score is determined and pruned away by the pruning procedure. This can cause memory thrashing and slow down the decoding speed.

In the *wave decoder*, a soft beam based on the existing arcs is calculated first. Arc predication is performed before a new arc is created. The arc predication engine first calculates the path score entering the first state of that arc before allocating that missing arc. The path score is compared with the current soft beam. If it is within the soft beam, the arc is created and the soft beam is updated. This scheme is admissible assuming that the selected beam width is appropriate, because it introduces a lower cut-off score than the cut-off score determined by the correct beam. This admissible arc predication scheme is useful for the dynamic decoder, but it makes almost no difference for a static decoder, because in that scenario, the decoding network is always fully built. For a dynamic decoder, this admissible arc predication scheme is quite effective in reducing the mortality rate of the newly created arcs, one of the critical problems which plague the efficiency of the dynamic decoder.

4. DYNAMIC NETWORK OPTIMIZATION

The use of detailed acoustic models in speech recognition makes it important to minimize the redundancy in the decoding network. In theory, for context dependent tri-phone modeling, there are $N \times M$ connections for each center phone in the DP net, where N is the number of different left phone context and M is the number of different right phone context. However, depending on the context dependent tri-phone model unit inventory in the acoustic model, many tri-phone model units are cloned using the available tri-phone model units which are most closely resemble the missing tri-phone model units. As a consequence, the number of distinct arc connections is often much less than $N \times M$. Optimizing a fully compiled static DP network is relatively straightforward, whereas reducing the redundancy in a dynamically constructed DP network is more involved. The situation becomes more acute when the network is not only constructed dynamically but being punctured due to various partial releases. In this situation, the local network is not fully built, and the redundancies if existing are hard to find. In the *wave decoder*, the network generation engine employs certain optimization procedures in creating the DP network.

In order to reduce arcs in dynamic network building, a model inventory hashing table for distinct tri-phone models in the existing arcs is maintained at the corresponding scaffolding phone node. For each node requiring an arc expansion, the following procedures are performed to minimize the dynamically generated decoding network.

- If there is an existing arc with the same tri-phone model unit from that node, no new arc is created and the arc is shared with a new destination node added for that arc.
- If there exists an existing arc with the same tri-phone model unit from a different DP node in the node hash table of the same phone node, no new arc is created. The arc is shared with a new initialization node added and the desired destination node is updated.
- If no existing arc with the desired tri-phone model unit, a new arc is created and inserted in the table.

These procedures are performed at the local $N \times M$ DP net level, and therefore, such arc sharing becomes possible.

In fact, the minimized self-adjusting local search graph is a generalized graph (not the graph in the strict sense), because one arc can have multiple starting nodes and multiple ending nodes as a result of incorporating the structure of acoustic model in search graph minimization.

Similar procedures can also be applied to the nodes of the DP network. If two nodes have identical non-punctured forward arc collections, two nodes are merged and redundant arc collections are released. However, this has to be done carefully, requiring two nodes at the same local segment being both active with non-punctured arc collections.

5. EXPERIMENTAL RESULTS

The proposed approach was applied in several applications. One application is the recognition of known length connected digit strings. If fully expanded, the decoding grammar has a depth of 16 levels. The acoustic model used in the experiments was a set of 274 context dependent acoustic model units[2]. The inter-word context dependency was explicitly modeled and each digit model was represented as a context dependent graph with 12 fan-in heads, one body and 12 fan-out tails. The speed overhead of maintaining a self-adjusting search graph in the *wave decoder* is negligible (<5%) and the peak decoding memory usage is only 6% of the corresponding static decoder. For multi-channel automatic speech recognition (ASR), additional savings can be achieved by sharing the scaffolding layer across different channels. The peak memory usage is determined mainly by the number of active channels which reaches the peak of their decoding memory usage. Because the peak memory usage is sparsely distributed over the duration of the speech input, the actual memory usage for a multi-channel decoder is even much less. We also tested the *wave decoder* for a most compact re-entrant grammar, a no-grammar network with a re-entrant null arc transition, using the same set of acoustic model. The total peak decoding memory usage is only 80KB (0.08MB) which is an order of magnitude reduction comparing to the static decoder. In fact, for a dynamic decoder, it is relatively easy to get significant savings on a huge network, whereas savings on a very compact grammar depend on the tightness of the self-adjusting search graph at each time instant.

We also tested the proposed approach on a task with 5.6K vocabulary words. In order to test decoder, each word was made distinctive (no phone tree sharing) with distinct 5.6K fan-out connections so that the search had a perplexity of 5.6K as well. The acoustic model used in the search contains 1769 context dependent tri-phone model units including context dependent one phone cross word units. A static decoding network required 250MB to build, while the peak decoding memory usage of the *wave decoder* was 10MB, a more than an order of magnitude reduction.

Task	*Ori*	*Wave*	
	(Mem)	(Mem)	(CPU Overhead)
Conn. Digit (NG)	0.8MB	0.08MB	< 2%
Known Length	2.5MB	0.1MB	<5%
LVR (PPLX=5623)	250MB	10MB	<15%

Table 1: Wave decoder vs. static decoder

6. SUMMARY

In this paper, a *wave decoder* based on the general re-entrant network for continuous speech recognition is described. The decoder design is based on the concept of self-adjusting decoding graph in which the decoding network is expanded and released frame-synchronously. The fast network expansion and release are made possible by utilizing a novel dynamic network scaffolding layer. The self-adjusting decoding graph is obtained by slicing the traditional decoding network horizontally for separation of different knowledge sources and vertically according to each time instant in search. A two layer hashing structure and an admissible arc predication scheme are described. These methods significantly reduce the arc mortality rate, a problem which plagues the efficiency of the dynamic decoder. In addition, deeper pruning is made possible in the proposed approach, which eliminates procedures of path halting and other incomplete erasures in a single pass dynamic decoder. Experimental results demonstrate that an order of magnitude reduction of decoding resources can be achieved based on the proposed approach.

REFERENCES

[1] Chin-Hui Lee and Lawrence R. Rabiner "A Frame-Synchronous Network Search Algorithm for Connected Word Recognition", *IEEE Trans. on Acoustic and Signal Processing*, vol 37, No. 11, November 1989.

[2] C.-H. Lee et al, "Context dependent acoustic modeling for connected digit recognition", 1993 ASA Fall meeting, Denver, Oct 93.

[3] J.J. Odell, V. Valtchev, P.C. Woodland and S.J. Young, "A One Pass Decoder Design for Large Vocabulary Recognition", *DARPA Work-Shop on Continuous Speech Recognition, March 94*, pp 380–385.

[4] E. Giachin, C.-H. Lee, R. Pieaccini and L.R. Rabiner, "Implementation Aspects of Large Vocabulary Recognition Based on Intra-Word and Inter-Word Phonetic Units", *DARPA Work-Shop on Continuous Speech Recognition, June 90.*

[5] F. Allieva, X.-H. and M.-Y. Hwang "An Improved Search Algorithm Using Incremental Knowledge for Continuous Speech Recognition", *Proc. ICASSP 93.*

[6] J. Murakami and S. Sagayama, "An Efficient Algorithm for Using Word Trigram Models for Continuous Speech Recognition," *Proc. of Fourth Australian International Conference on Speech Science and Technology*, pp 330–335, 1992.

[7] D. B. Paul, "An Efficient A^* Stack Decoder Algorithm for Continuous Speech Recognition with a Stochastic Language Model", *Proc. ICASSP 92.*

[8] N.J. Nelsson "Principles of Artificial Intelligence", *McGraw Hill*, 1980.

[9] H. Ney, "Architecture and Search Strategies for Large-Vocabulary Continuous Speech Recognition", *pp 59 – 84* NATO-ASI BUBION 93.

[10] J. Pearl "Heuristics", *Addison and Weiley 1984*