A DYNAMIC NETWORK DECODER DESIGN FOR LARGE VOCABULARY SPEECH RECOGNITION

V. Valtchev, J.J. Odell, P.C. Woodland, & S.J. Young

Cambridge University Engineering Department,
Trumpington Street, Cambridge, CB2 1PZ, England.

ABSTRACT

Accuracy and speed are the main issues to consider when designing a large vocabulary speech recogniser. Recent experience with the Wall Street Journal (WSJ) corpus [5], has shown that high recognition accuracy requires the use of detailed acoustic models in conjunction with well-trained long span language models. In this paper we present a two-pass decoder architecture which extends an original [4] one-pass design. The initial pass consists of a time synchronous backward search in a pre-compiled network using simplified acoustic models and a null grammar. The forward pass can function as a stand-alone one-pass decoder capable of using cross-word context-dependent models and long span language models. The capabilities of this framework are empirically examined in terms of recognition accuracy vs speed on the Wall Street Journal database.

1. INTRODUCTION

Hidden Markov Models (HMMs) constitute the most successful approach to automatic speech recognition. Part of the success of the HMM framework is the existence of an automatic supervised training algorithm with mathematically proven convergence (the Baum-Welch algorithm) and an efficient decoding scheme used in recognition of unknown utterances (the Viterbi algorithm). The conventional approach to the recognition (decoding) of unknown speech utterances is to apply the Viterbi search algorithm to a pre-compiled network of HMM instances [6]. The representational ability of such static networks makes them well-suited to small to medium vocabulary tasks. However, as recognition tasks become more complex, the size of the static network needed grows dramatically especially if cross-word context is included and longer span language models are used.

Decoding speech by simultaneously applying the best knowledge sources is likely to result in more efficient pruning and achieve higher recognition accuracy. In this paper we compare the performance of a one-pass decoder architecture and a multi-pass design. The structure of the single pass decoder allows for the easy incorporation of lookahead information with a view to further minimising the search effort. As such, the multi-pass design can be built by extending the original one-pass decoder to incorporate a preliminary backward pass.

The paper proceeds with the discussion of conventional Viterbi decoding using static networks, its limitations and possible ways of improving performance. The design of a one-pass decoder architecture using a dynamically constructed tree-structured network is then presented. Section 4 outlines a multi-pass approach that can be used to improve the performance of the one-pass decoder. This is followed by implementational details of the backward search used as a preliminary pass in the multi-pass decoder. Finally, experimental results are presented showing the performance of the one-pass decoder and the multi-pass approach.

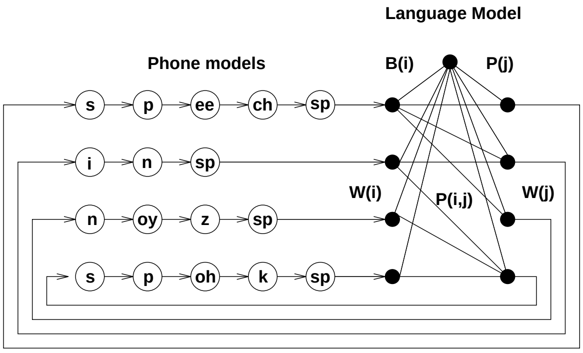

Figure 1. Static network for Viterbi decoding with a bigram back-off language model

2. VITERBI DECODING

The standard method of implementing a Viterbi search for decoding an unknown speech utterance into words is to build a re-entrant network of HMM phone instances (see Fig. 1). Words in the vocabulary are constructed as a sequence of phones linked according to the word's pronunciation rules. Decoding proceeds as a time-synchronous Viterbi search through this pre-assembled network. In the work presented here, recognition is performed using the token passing paradigm [7] where the best path to a particular instance in the network is described by a token held in that state. Partial paths are extended by matching input frames against the phone models and propagating tokens with updated scores into viable successors.

2.1. Network Structure

Current state-of-the-art systems use cross-word context-dependent models and stochastic long span language models. Incorporating such detailed knowledge sources into the recognition process has begun to show the representational constraints imposed by static networks (Fig.1). For example, the use of a trigram language model requires paths of different two word histories to be maintained separately. For a large vocabulary size static representation of such a structure may prove impossible. Similarly, allowing for cross-word context dependencies can lead to a substantial increase in the size of the network.

Since the uncertainty in decoding speech is much higher at the start of words than at ends, it follows that the majority of the effort is spent on decoding the first few phones of each word [3]. For large vocabulary tasks, efficiency can be improved by building a tree structured network in which words with common initial phone sequences share corresponding acoustic models. Although such a representation can be modelled as a re-entrant structure, this will not easily allow for the use of cross-word triphones or long span language models.

2.2. Beam Search

The standard time-synchronous Viterbi search employs a breadth-first strategy where multiple paths are extended in parallel by matching each input frame in turn. The number of active phone instances for each time frame is the most indicative measure of the computational effort required to explore the present search space. Beam search is the standard technique used to reduce the search during decoding. In practical systems, variable width and multiple beam schemes have proven more effective compared to a single fixed threshold [2]. For example, the following types of pruning are commonly found in current state-of-the-art decoders.

- *Model pruning* threshold used to deactivate models outside the beam from the current best path.
- *Word-end pruning* beam used to restrict token propagation out of word-end nodes. This is justified on the basis that a much higher degree of uncertainty exists at the start of words than at the end. This is further supported by the limited dynamic range of the language model probabilities when leaving a word, due to the heavy reliance on the back-off component of the language model.
- *Variable width* pruning is a way of improving performance based on a global measure such as the total number of active phone models. This has the effect of automatically reducing the beamwidth once the limit is reached.

3. ONE-PASS DECODER

The key features of a successful one-pass decoder are the ability to incorporate cross-word models and long span language models whilst keeping storage and computational requirements low. To do this it is necessary to tree structure the recognition network and to apply tight and efficient pruning. In order to allow for cross-word context and long span language models, the decoder must be able to obtain context at word and phone level in an efficient way. To make such context explicit and easily available one has to unfold the static network structure. To limit the storage requirements for building such a structure, the network must be grown *on-the-fly* and once instances fall outside of the beam, the corresponding nodes must be reclaimed.

Due to the tree structuring of the network and the possibility that two words may have exactly the same phonetic transcription (and consequently share the same path of models), it is necessary to introduce some point where the identity of the word becomes unique. This will also facilitate the process of combining language and acoustic scores and allow for an explicit application of word-end style pruning. Consequently the recognition network consists of two types of nodes.

- *Word internal nodes* - these nodes are linked together according to the phonetic transcription of the words they represent. Such nodes have an associated HMM instance chosen according to surrounding phone/word context. The HMM instance is used to compute the likelihood of the current input frame and to hold tokens representing paths.
- *Word-end nodes* - these nodes uniquely identify each word in the lexicon. Language model scores are added during token propagation into word-end nodes and the combined acoustic/language model score is used for word-end pruning. When operating as the second pass of a multi-pass decoder these nodes will have associated bitmaps keeping track of started word followers.

Each node in the network has an associated language model probability. This is added to the token likelihood to give a total score used for pruning. At *word-end* nodes, the language model score is the exact probability for the ending word given its predecessors. However, for *word internal* nodes shared by more than one word the language model score is an exact upper bound (the maximum language model probability of all words sharing that node). Using the exact upper bound allows for the application of tight pruning without any adverse effects on accuracy. No reliance is made on the back-off nature of the language models so the computational load will not increase as the size of the language model grows. The decoder operates using the following algorithm:

```
create root node;
for each input frame {
    compute frame likelihood;
    propagate phone-internal tokens;
    do model pruning;
    propagate word-internal tokens;
    do word-end pruning;
    propagate word-external tokens;
    prune network;
}
recover word sequence from best sentence-end node;
```

The network is constructed on the fly and nodes are created only when they fall into the beam. Network growth occurs during *word-internal* token propagation when a token is to be propagated from a node without a follower. New nodes can also be created during *word-external* token propagation. In this case, a full new tree commences growing. Network pruning maintains a minimal network structure and can be summarised as follows.

- *Forward pruning* - nodes outside the beam without predecessors will be permanently removed from the network and associated storage will be reclaimed. This procedure is performed by scanning network nodes in a forward fashion.
- *Backward pruning* - nodes outside the beam without followers will be removed and associated storage reclaimed. The predecessor node will be updated to contain information sufficient to recreate the node if it falls back into the beam. This procedure is performed by traversing the network in a backward fashion.

A maximum model pruning algorithm is also incorporated, whereby the structure of the network is examined before token propagation takes place and a rough estimate is obtained of the potential network growth. If this exceeds the maximum number of active phone instances allowed, the pruning threshold

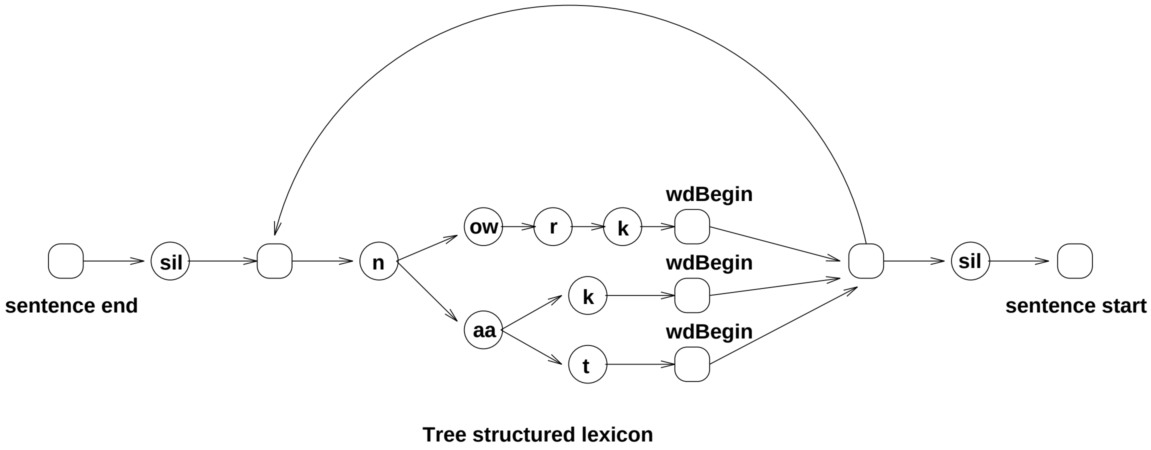

Figure 2. Static tree structured network used in the backward pass

is automatically readjusted in order to reposition the growth within the set limit.

4. THE BACKWARD-FORWARD SEARCH

A variety of schemes have been proposed aimed at reducing the computational effort required during recognition. One approach attempts to speed up recognition together with reducing storage by decomposing the recognition process into several less complex stages. Reduction in storage is achieved by progressively applying more sophisticated knowledge sources (acoustic/language models) as the recognition moves from one stage to another [1]. Originally, it was thought that using all available knowledge sources in a single pass would be too costly to perform. However, our recent work in [4] has shown that this is not necessarily the case.

In this section we discuss possible ways of further improving the performance of the one-pass decoder. In the one-pass decoder a new copy of the lexicon tree is grown for each word-end node within the word-end pruning beam. The tree structuring mechanism ensures that the number of immediate successors of an active word-end node is relatively small even for the cross-word context case. Since the search effort in the recogniser is dependent upon the number of active HMM instances and since no prior knowledge is available about the utterance beyond the current input frame, many models will be made active in order to hypothesise all words as possible successors. Furthermore, even with tree structuring, a larger lexicon will generate more dense trees with a potentially high number of active phone instances. Many of these models will fall out of the beam shortly after activation without any contribution to the overall recognition of the utterance. Words ending with a good acoustic score will always have to be expanded even when they constitute a path which is not acoustically/grammatically consistent with the remaining part of the utterance.

A two stage search has been implemented in an attempt to solve these problems. The search is similar to the forward-backward search in [1], but it runs the first pass in the backward direction, with a second forward pass. In the simplest case, the backward pass is used to construct a list of active word-begin nodes for every time frame in the utterance. This list can then be used during the main forward pass to control the growth of new trees at surviving word-end nodes.

5. THE BACKWARD SEARCH

The backward search is built as a backward time-synchronous Viterbi pass using a pre-compiled network of phone instances. The network (Fig. 2) is automatically generated at start up time from the pronunciations in the dictionary. Since the aim is to provide acoustic lookahead information, no language model is used. This allows for tree-structuring the network and thus significantly reducing computation. Silence is enforced at the start and end of each utterance. The output of the backward pass consists of a list of words and associated scores for each time frame of the input utterance. Separate pruning thresholds are used to control the activation of word-internal nodes and word-begin nodes. The word-begin pruning beamwidth in conjunction with a top-N selection scheme are used to adjust the number of words kept for each time frame.

There are two issues that need to be considered [2]. The first issue is how accurate the first pass should be. For example, using simplified acoustic models the search will run very quickly, however, the word lists will be either too long to constrain the second pass, or in the case of tight pruning, the word lists will have too many missing words resulting in unrecoverable search errors. Rough acoustic models will require less effort in the HMM's output probability calculation, however, pruning will be less efficient due to larger variance. On the other hand, a search using more detailed acoustic models (e.g. word-internal triphones) will take longer to perform, resulting in shorter word lists even for wide pruning beamwidths. The second issue is how to use the information (scores and word id's) obtained in the backward pass. In order to incorporate the acoustic lookahead information, the forward pass needs to keep track of how many words have already been started from every word-end node in the network. This is achieved by associating bitmaps of word identifiers with each word-end node. Information from the backward pass can be used in two ways. In the first case, the list of words is compared with the bitmap of the word-end node under consideration. Any words not already present are incorporated into the existing tree of followers. This avoids the use of the somewhat unreliable acoustic scores generated by the simplified acoustic models. In the second case the word-end pruning strategy can be performed according to the measure [1]

$$\frac{\alpha(w,t)\beta(w,t)}{\beta^T} \qquad (1)$$

Backward	Type	Error %	Time
none	*none*	12.8	13.2
160/120	top 80 words	13.5	5.5
160/120	top 200 words	13.2	8.2
160/120	all words	12.9	11.2
160/80	B/F pruning	13.5	4.5
160/100	B/F pruning	13.0	5.2
160/120	B/F pruning	12.9	5.5
180/150	B/F pruning	12.8	6.6

Table 1. Nov 1993, WSJ 5K closed vocabulary test results using 8 mixture monophones in the backward pass and 8 mix word-internal triphones in the forward pass with bigram back-off language model. Time is given in minutes per sentence, and the numbers in format 160/120 give the HMM pruning threshold and the word-end pruning threshold respectively for the backward pass.

Backward	LM	Error %	Time
none	bigram	8.8	21.5
180/150	bigram	9.1	12.3
none	trigram	6.9	19.3
180/150	trigram	7.5	12.0

Table 2. Nov 1993, WSJ 5K closed vocabulary test results using 8 mixture monophones in the backward pass and 8 mixture gender dependent cross-word triphone models in the forward pass.

where $\beta(w,t)$ is the backward Viterbi likelihood associated with the beginning of word w at time t, β^T is the likelihood of the utterance as computed in the backward pass and $\alpha(w,t)$ is the forward Viterbi likelihood associated with the word-end node for w at time t. The above pruning measure will result in word-end nodes with higher likelihood scores starting larger trees of followers than those with lower likelihoods.

6. RESULTS & DISCUSSION

The set of experiments presented here attempts to establish the usefulness of the multi-pass approach when compared to the original one-pass design. The multiple pruning schemes employed in the two passes allow for a variety of experiments to be performed. The results presented in this paper are concerned with the problem of achieving the highest possible accuracy with minimal search effort i.e. no attempt has been made to establish the performance of the two systems when running at very low pruning beamwidths as may be required in real time applications.

Experiments have been performed on the November 1993, 5K evaluation test data from the Wall Street Journal task. The systems used training data from the SI-84 data sets and pronunciations from the Dragon Wall Street Journal Pronunciation Lexicon Version 2.0. Standard bigram and trigram back-off language models were used as supplied by MIT Lincoln Labs. The models were built using the HTK Hidden Markov Model toolkit. The system employed models with three emitting states, left-to-right topology and continuous density mixture Gaussian output distributions tied at the state level using phonetic decision trees. As described above, both passes enforce silence at the start and end of each utterance and allowed optional silence between words. All experiments were performed on a Silicon Graphics Indigo R4000 workstation.

Table 1 presents results demonstrating the use of simple word lists vs backward-forward pruning to constrain the forward pass. These show that word lists without acoustic scores are too coarse to preserve the accuracy of the one-pass decoder at higher speeds. When the acoustic likelihood from the backward pass is used to derive the word-end pruning threshold in the forward pass, 2 - 3 times improvements in speed can be observed and minimal or no loss in accuracy. The results in Table 2 show that the speed improvement obtained using the backward pass reduces as the accuracy of the forward pass increases. This can be attributed to more efficient pruning as a result of more precise acoustic/language modelling. The results also suggest that using more detailed models in the forward pass will require further tuning of the backward pass.

7. CONCLUSIONS

The multi-pass decoder architecture can provide a speed up over the one-pass design. However, the relative improvements in speed decrease as the complexity of the models used in the forward (detailed) pass increases. Furthermore, improved acoustic/language models in the forward pass are likely to require better models in the backward pass which could offset the speed gains even further. In general, the performance of the multi-pass approach depends to a large extent on the fine tuning of the backward pass. Experiments are under way to establish the performance of the two designs with a view to real-time system applications.

8. ACKNOWLEDGEMENTS

The WSJ pronunciation dictionary was provided by Dragon Systems Inc. J. Odell is funded by a EPSRC studentship and part of this work was funded by EPSRC grant GR/J10204.

REFERENCES

[1] S. Austin, R. Schwartz, and P. Placeway. The Forward-Backward Search Algorithm. In *Proc. ICASSP'91*, volume 1, pages 697–700. IEEE, May 1991.

[2] R. Lacouture and Y. Normandin. Efficient Lexical Access Strategies. In *Proc. Eurospeech'93*, Berlin, 1993.

[3] H. Ney, R. Naeb-Umbach, B-H. Tran, and M. Oerder. Improvements in Beam Search for 10000-Word Continuous Speech Recognition. In *Proc. ICASSP'92*. IEEE, 1992.

[4] J.J. Odell, V. Valtchev, P.C. Woodland, and S.J. Young. A One Pass Decoder Design For Large Vocabulary Recognition. In *Proc. ARPA Human Language Technology Workshop*. ARPA, March 1994.

[5] P.C. Woodland, J.J. Odell, V. Valtchev, and S.J. Young. Large Vocabulary Continuous Speech Recognition Using HTK. In *Proc. ICASSP'94*, Adelaide, March 1994. IEEE.

[6] S.J. Young. The HTK Hidden Markov Model Toolkit: Design and Philosophy. Technical Report TR.152, Cambridge University Engineering Department, 1993.

[7] S.J. Young, N.H. Russell, and J.H.S. Thornton. Token Passing: a Simple Conceptual Model for Connected Speech Recognition Systems. Technical Report TR.38, Cambridge University Engineering Department, July 1989.