# Integrated Context-Dependent Networks in Very Large Vocabulary Speech Recognition

*Mehryar Mohri* mohri@research.att.com

*Michael Riley* riley@research.att.com

AT&T Labs – Research, 180 Park Avenue, Florham Park, NJ 07932-0971, USA

**Abstract**

All the components used in the search stage of speech recognition systems – language model, pronunciation dictionary, context-dependent network, HMM model – can be represented by finite-state labeled networks. To construct real-time recognition systems, it is important to optimize these networks and to efficiently combine them. We present new methods that substantially improve these steps. We show that an efficient recognition network including context-dependent and HMM models can be built using weighted determinization of transducers [6]. We report experiments with a 463,331-word vocabulary North American Business News Task that show a substantial improvement of the recognition speed over our previous method [9]. Furthermore, the size of the integrated context-dependent networks constructed can be dramatically reduced using a factoring algorithm that we briefly describe. With our construction, the integrated NAB network contains only about $1.3$ times as many arcs as the language model it is constructed from.

## 1. Introduction

All the components used in the search stage of speech recognition systems – language model, pronunciation dictionary, context-dependent network, HMM model – can be represented by finite-state labeled networks [9]. To construct real-time recognition systems, it is important to optimize these networks and to efficiently combine them. We outline new results that substantially improve these steps.

Optimization of these networks is crucial. Indeed, these networks are often highly *non-deterministic* – at a given state there might be several thousand alternative outgoing arcs, many of them labeled with the same input label. This redundancy directly affects the performance of the search. The problem has been addressed by some authors using *lexical trees* or other so-called *tree-based* representations [1, 4, 10, 11]. We have presented optimal methods based on weighted transducer determinization for eliminating such redundancies to address this problem [5, 6, 8].

It is also important to reduce the time required to combine these networks. In the best case, the combination work could be done beforehand, thus reducing to zero its contribution to the total search.

In previous work, we have shown that using weighted transducer determinization, one could optimize the networks corresponding to the language model, the pronunciation dictionary, and the context-dependency model and pre-combine them into a single labeled network whose size was only about twice that of the language model [9].

We present a new method that substantially improves the performance of our previous system by additionally incorporating the HMM state-level network into our optimization. We show that *all* components of a very-large vocabulary speech recognition system used in the search stage, including the acoustic model, can be pre-combined to build an integrated recognition network thus eliminating the combination time during the search. We further show that, using additional disambiguation symbols, this network can be *determinized* – at each state there is at most one outgoing arc labeled with any given input label – thereby eliminating the redundancy mentioned earlier.

To construct this optimized integrated network, we use weighted transducer determinization at each step of the composition of each two networks. We also use a new *factoring* algorithm that considerably reduces the size of the resulting integrated network.

Our experiments in the North American Business News task (NAB) with a 463,331-word vocabulary trigram language model show that the resulting integrated recognition network is more than two times smaller than the one obtained in our previous system, and its word accuracy is about $2.2\%$ better (in absolute value) at one times real-time on a SGI R10000.

## 2. Components of a speech recognition system

All the components of a speech recognition system can be represented by *weighted finite-state transducers* [2, 3], that is finite-state networks labeled with an input symbol, an output symbol, and a weight (typically the negative log of some probability). The symbol $\epsilon$ denotes the empty string, or the null symbol, its concatenation with any input or output sequence does not modify that sequence. A path in a weighted transducer pairs the concatenation of the input labels of its transitions with that of the corresponding output labels, assigning the pair the sum of the transitions weights. A weighted automaton is a weighted transducer which has identical input and output labels for any transition. The representation of each component by a weighted finite-state transducer is illustrated by figures 1 (a)-(d).

Figure 1 (a) shows a three-state HMM transducer mapping sequences of distribution indices to context-dependent phones, where our distributions are Gaussian mixture components. The global HMM transducer denoted by $H$ is obtained by taking the closure of the union of all HMMs used in acoustic modeling. Since our decoder directly simulates the self-loops of the HMM, they will be omitted in the following presentation.

Figure 1 (b) depicts a simple triphonic context-dependency transducer $C$ mapping context-dependent phones to phones with only two phones $d$ and $t$ [12]. Each state $(x, y)$ encodes the most recent pair of phones read. $e$ represents the start or end of a phone sequence.

A sample pronunciation dictionary transducer mapping phonemic transcriptions to word sequences is shown on figure 1 (c).

Figure 1 (d) is a weighted automaton representing a toy language model. Any $n$-gram language model can be represented in a similar way.

Weighted transducers map input sequences to output sequences with some weights. They can be composed just as other map-

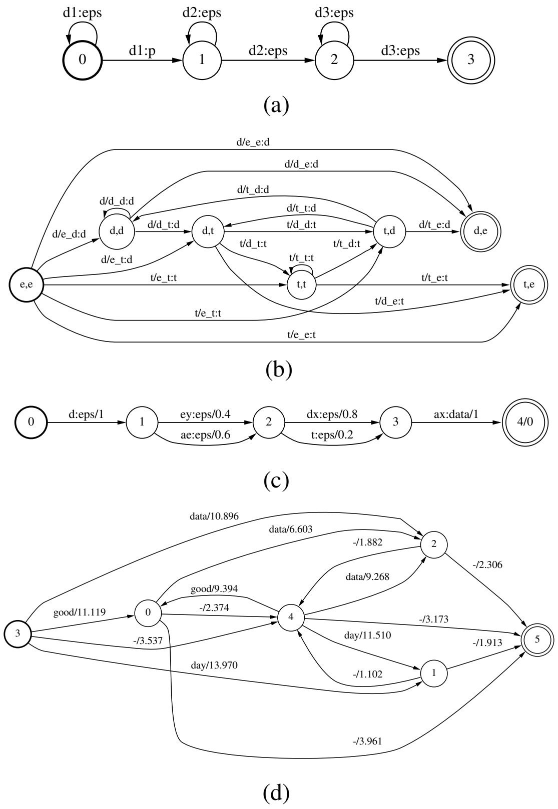


Figure 1: Weighted finite-state transducer representation of the components of a speech recognizer.

pings to create more complex mappings. In our case, the composition of the components:

$$H \circ C \circ L \circ G$$

gives a mapping from sequences of distribution names to word sequences. Figure 2 illustrates that recognition cascade. There exists a natural and efficient composition algorithm for combining weighted transducers [2, 7]. The algorithm also admits a natural on-the-fly implementation.

In previous work, we demonstrated that $C \circ L \circ G$ can be pre-constructed efficiently [9]. We showed that with that construction, the resulting network contains only about twice as many arcs as the corresponding language model in the NAB task. The next section describes a method for pre-constructing and optimizing $H \circ C \circ L \circ G$.

### 3. Network construction and optimization

To construct this optimized integrated network, we use weighted transducer determinization at each step of the composition of each pair of networks.

The main purpose of the use of determinization is to eliminate non-determinism in the resulting network, thereby substantially reducing recognition time. In addition, its use at intermediate steps of the construction helps to improve the efficiency of composition and to reduce the size of the networks.

It can be proved that in general, the transducer $L \circ G$ mapping phone sequences to words is not determinizable. This is clear in presence of homophones. But even in the absence of homophones, the unbounded delay imposed by the transduction makes $L \circ G$ non-determinizable: in some cases, it is not possible to determine even the first element of the word sequence corresponding to a phone sequence before reaching the end of that phone sequence [see [6] for a characterization of determinizable weighted automata and transducers].

To make it possible to determinize $L \circ G$, we introduce an auxiliary phone symbol denoted $\#_0$ marking the end of the phonemic transcription of each word. Other auxiliary symbols $\#_1 \ldots \#_{k-1}$ are used when necessary to distinguish homophones as in the following example:

| | |
|---|---|
| r eh d $\#_0$ | *read* |
| r eh d $\#_1$ | *red* |

At most $P$ auxiliary phones, where $P$ is the maximum degree of homophony, are introduced. The pronunciation dictionary transducer augmented with these auxiliary symbols is denoted by $\tilde{L}$. For consistency, the context-dependency transducer $C$ must also accept all paths containing these new symbols.

For further determinizations at the context-dependent phone level and distribution level, each auxiliary phone must be mapped to a distinct context-dependent phone. Thus, self-loops are added at each state of $C$ mapping each auxiliary phone to a new auxiliary context-dependent phone. The augmented context-dependency transducer is denoted by $\tilde{C}$.

Similarly, each auxiliary context-dependent phone must be mapped to a new distinct distribution name. $P$ self-loops are added at the initial state of $H$ with auxiliary distribution name input labels and auxiliary context-dependency output labels to allow for this mapping. The modified HMM model is denoted by $\tilde{H}$.

It can be shown that the use of the auxiliary symbols guarantees the determinizability of the transducer obtained after each composition. Weighted transducer determinization is used indeed at several steps of our construction.

To begin with, $\tilde{L}$ is composed with $G$ and determinized: [1] $det(\tilde{L} \circ G)$. The benefit of this determinization is the reduction of the number of alternative transitions at each state to at most the number of distinct phones at that state, while the original network may have as many as $V$ outgoing transitions at some states where $V$ is the vocabulary size. For large tasks where the vocabulary size can be more than several hundred thousand, the advantage of this optimization is clear.

The inverse of the context-dependency transducer might not be deterministic. [2] For example, the inverse of the transducer shown in figure 1 (c) is not deterministic since the initial state admits several outgoing transitions with the same input label $d$ or $t$. To build a small and efficient integrated network, it is important to first determinize the inverse of $C$. [3]

$\tilde{C}$ is then composed with the resulting transducer and determinized. Similarly $\tilde{H}$ is composed with the context-dependent network and determinized. This last determinization increases sharing among HMM models that start with the same distributions: at each state of the resulting integrated network, there is at most one outgoing transition labeled with any given distribution name. This leads to a substantial reduction of the recognition time.

[1] An $n$-gram language model $G$ is often constructed as a deterministic weighted automaton with a back-off state – in this context, the symbol $\epsilon$ is treated as a regular symbol for the definition of determinism. If this does not hold, $G$ is first determinized [6].

[2] The inverse of a transducer is the transducer obtained by exchanging input and output labels of all transitions.

[3] Triphonic or more generally $n$-phonic context-dependency models can be built directly with a deterministic inverse [12].

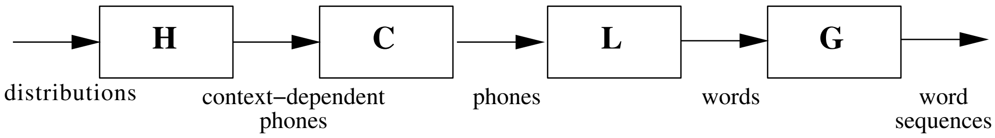


Figure 2: Recognition cascade.

As a final step, the auxiliary distribution symbols of the resulting network are simply replaced by $\epsilon$'s. The corresponding operation is denoted by $\pi_\epsilon$. The sequence of operations just described is summarized by the following construction formula:

$$N = \pi_\epsilon(det(\tilde{H} \circ det(\tilde{C} \circ det(\tilde{L} \circ G))))$$

where parentheses indicate the order in which the operations are performed. The result $N$ is an integrated recognition network that can be constructed even in very large-vocabulary tasks and leads to a substantial reduction of the recognition time as shown by our experimental results. The next section presents a method for reducing the size of the network while preserving its efficiency.

## 4. Factoring

Our decoder has a separate represention for variable-length left-to-right HMMs for both time and space efficiencies, which we will call the *HMM specification*. However, the integrated network of the previous does not take good advantage of this since, having combined the HMMs into the recognition network proper, the HMM specification consists of trivial one-state HMMs. However, by suitably *factoring* the integrated network, we can again take good advantage of this feature.

A path whose states other than the origin and the destination have no more than one outgoing or incoming transition is called a *linear path*. The integrated recognition network just described may contain many linear paths after the composition with $\tilde{H}$, and after determinization. The set of all linear paths of $N$ is denoted by Lin$(N)$.

Input labels of $N$ name one-state HMMs. We can replace the input of each linear path of $N$ of length $n$ by a single label naming an $n$-state HMM. The same label is used for linear paths having the same input. The result of that replacement is a more compact transducer denoted by $F$. The factoring operation on $N$ leads to the following de-composition:

$$N = H' \circ F$$

where $H'$ is a transducer mapping variable-length left-to-right HMM names to $n$-state HMMs. Since $H'$ can be separately represented via the decoder's HMM specification, the actual recognition network is reduced to $F$.

Linear paths inputs are in fact replaced by a single label only when this helps reducing the size of the network. This can be measured by defining the *gain* of the replacement of an input sequence $\sigma$ of a linear path by:

$$G(\sigma) = \sum_{\pi \in \mathrm{Lin}(N), i[\pi]=\sigma} |\sigma| - |o[\pi]| - 1$$

where $|\sigma|$ denotes the length of the sequence $\sigma$, $i[\pi]$ the input label and $o[\pi]$ the output label of a path $\pi$. The replacement of a sequence $\sigma$ helps reducing the size of the network if $G(\sigma) > 0$.

Our implementation of the factoring algorithm allows one to specify the maximum number $r$ of replacements done (the $r$ sequences with the highest gain are replaced), as well as the maximum length of the linear chains considered.

Factoring does not affect recognition time, however it leads to a substantial reduction of the size of the network as will be shown in the next section.

## 5. Experimental results

We used the techniques outlined in previous sections to construct an integrated optimized recognition network for a 463,331-word vocabulary North American Business task (463,331 represents the total number of words found in the corpus).

To measure the improvement from these techniques, we compared the size of our recognition network and its recognition performance to the one obtained using our previous construction [9]. The same acoustic, lexical, and language models were used in these experiments:

- The acoustic model was 5,520 distinct HMM states, each associated to a four-Gaussian mixture model.
- $C$ represented a triphonic context-dependent model with 1,523 states and 80,719 transitions.
- $L$ was a 463,331-word pronunciation dictionary.
- $G$ was a trigram language model with 5,285,995 transitions built using Katz's backoff method with frequency cutoffs of 2 for bigrams and 4 for trigrams and shrunk with an epsilon of 10 using the method of Seymore and Rosenfeld [13].

We used the factoring algorithm described in the previous section to reduce the size of unfactored network $N$ built from these components. Table 1 gives the size of the integrated recognition network $N = H \circ C \circ L \circ G$. Factoring helped reduce the size of the network by about 4 times without affecting recognition speed or accuracy. The comparison with the context-dependent network $C \circ L \circ G$ constructed with our previous method [9] shows that our new recognition network is about 2 times smaller.

Observe that the size of the new integrated factored network $F$ is close to that of the language model used: $F$ has only about 30% more transitions than $G$. The HMM specification $H'$ consists of 613,440 HMMs with an average of 10.7 states per HMM. It occupies only about 20% of the memory of $F$ in the decoder (due to the compact representation possible from its specialized topology) and thus the overall memory reduction from our factoring is substantial.

Table 1: Size of recognition networks in NAB 463,331-word vocabulary task.

| network | states | transitions |
|---|---|---|
| $C \circ L \circ G$ | 9,732,230 | 13,415,569 |
| $H \circ C \circ L \circ G$ | 23,553,133 | 27,240,013 |
| $F$ | 3,215,515 | 6,902,395 |

Our experiments with the NAB 463,331-word vocabulary task using a simple general-purpose one-pass Viterbi decoder show

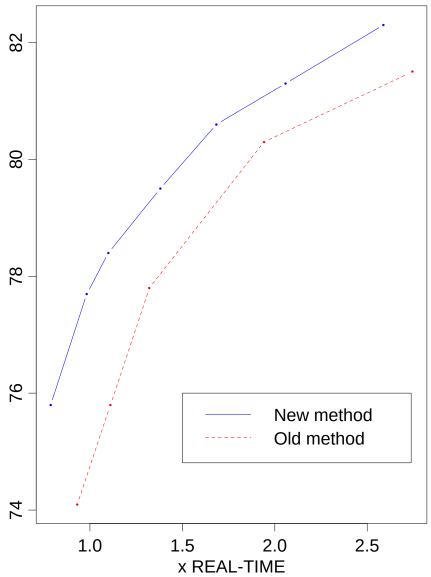


Figure 3: Comparison of the old method based on the construction of the network $C \circ L \circ G$ versus the new method based on the deterministic construction of $H \circ C \circ L \circ G$ in the 463,331-word vocabulary NAB task.

that the integrated recognition network $N$ (or $F$) described in the previous sections substantially speeds up recognition.

Figure 3 gives recognition accuracy as a function of recognition time, in multiples of real time on a single processor of a Silicon Graphics Origin 2000 for the network constructed using our old method and for our new integrated recognition network.

With our new construction method, the real-time accuracy is improved by more than $2.2\%$ in absolute value. The accuracy achieved by the new integrated network at real-time is only reached by the old system at about $1.35\%$ times real-time.

Interestingly, we found that with our new construction, the benefit of the use of trigram language models versus bigram models was clear even at $.5$ times real-time.

Figure 4 shows recognition results for trigram models in comparison with results for bigrams for the same vocabulary size, 20,000, in the NAB task. The context-dependency model and the acoustic model used were the same as those described above for the 463,331-word vocabulary task. The word accuracy of the network constructed from a trigram language model is always significantly better than that of the bigram model in the range of interest.

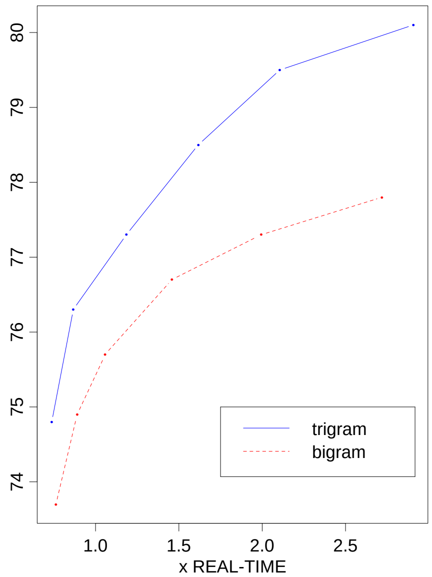


Figure 4: Comparison of the use of a bigram versus a trigram language model in the $20,000$-word vocabulary NAB task.

## 6. Conclusion

A general method for constructing efficient integrated networks including context-dependent and HMM models was described. The method, based on the use of weighted determinization and a new factoring algorithm, provides recognition networks that are practical and that can be used to build a 463,331-word vocabulary single-pass speaker-independent real-time speech recognition system in the NAB task.

The integrated recognition networks constructed are significantly more efficient, both in space and time, than the networks constructed using our previous method [9].

## 7. References


1. G. Antoniol, F. Brugnara, M. Cettolo, and M. Federico. Language model representations for beam-search decoding. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '95)*, pages 588–591, Detroit, MI, 1995.
2. J. Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher: Stuttgart, 1979.
3. S. Eilenberg. *Automata, Languages and Machines*, volume A-B. Academic Press, 1974-1976.
4. P. S. Gopalakrishnan, L. R. Bahl, and R. L. Mercer. A tree search strategy for large-vocabulary continuous speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '95)*, pages 572–575, Detroit, MI, 1995.
5. M. Mohri. On some applications of finite-state automata theory to natural language processing. *Journal of Natural Language Engineering*, 2:1–20, 1996.
6. M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23:2, 1997.
7. M. Mohri, F. C. N. Pereira, and M. Riley. Weighted automata in text and speech processing. In *ECAI-96 Workshop, Budapest, Hungary*. ECAI, 1996.
8. M. Mohri and M. Riley. Network optimizations for large vocabulary speech recognition. *Speech Communication*, 25:3, 1998.
9. M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. C. N. Pereira. Full expansion of context-dependent networks in large vocabulary speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP '98)*, Seattle, Washington, 1998.
10. J. Odell, V. Valtchev, P. Woodland, and S. Young. A one pass decoder design for large vocabulary recognition. In *Proceedings of the ARPA Human Language Technology Workshop, March 1994*, pages 405–410, 1994.
11. S. Ortmanns, H. Ney, and A. Eiden. Language-model look-ahead for large vocabulary speech recognition. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP'96)*, pages 2095–2098. University of Delaware and Alfred I. duPont Institute, 1996.
12. M. Riley, F. C. N. Pereira, and M. Mohri. Transducer composition for context-dependent network expansion. In *Proceedings of Eurospeech'97*. Rhodes, Greece, 1997.
13. K. Seymore and R. Rosenfeld. Scalable backoff language models. In *Proceedings of ICSLP*, Philadelphia, Pennsylvania, 1996.