FULL EXPANSION OF CONTEXT-DEPENDENT NETWORKS IN LARGE VOCABULARY SPEECH RECOGNITION

Mehryar Mohri *Michael Riley* *Don Hindle* *Andrej Ljolje* *Fernando Pereira*

AT&T Labs – Research
180 Park Avenue, Florham Park, NJ 07932, USA
{mohri, riley, dmh, alj, pereira}@research.att.com

ABSTRACT

We combine our earlier approach to context-dependent network representation with our algorithm for determinizing weighted networks to build optimized networks for large-vocabulary speech recognition combining an n-gram language model, a pronunciation dictionary and context-dependency modeling. While fully-expanded networks have been used before in restrictive settings (medium vocabulary or no cross-word contexts), we demonstrate that our network determinization method makes it practical to use fully-expanded networks also in large-vocabulary recognition with full cross-word context modeling. For the DARPA North American Business News task (NAB), we give network sizes and recognition speeds and accuracies using bigram and trigram grammars with vocabulary sizes ranging from 10,000 to 160,000 words. With our construction, the fully-expanded NAB context-dependent networks contain only about twice as many arcs as the corresponding language models. Interestingly, we also find that, with these networks, *real-time* word accuracy is improved by increasing vocabulary size and n-gram order.

1. INTRODUCTION

In previous work [9, 13, 15] we have shown that weighted automata provide a competitive unifying framework for all stages of network creation and combination in speech recognition. In this framework, a single algorithm, *transducer composition* [1], is used to combine the input acoustic observations and various modeling networks: acoustic models, the context-dependency model, pronunciation dictionary and language model.

It is well accepted that context-dependent phone models are very useful in high-accuracy recognition [5, 17]. However, given the size of the various models in large-vocabulary speech recognition, we might expect that the fully-expanded context-dependent model network built by combining cross-word context-dependent models with a pronunciation dictionary and an n-gram language model would be too big to be stored or used in an efficient recognizer.

In applications with dynamically-changing language models or dictionaries, it is not possible to build the full modeling network in advance, so dynamic network composition is needed, such as, for instance, our on-demand network composition method [15]. In applications with fixed language model and dictionary, however, there is in principle the opportunity for combining all the modeling networks in advance into an optimized network. Whether this is also practical depends crucially on the optimization methods used to avoid state explosion in cross-word settings, as we shall describe presently. In particular, we will show that fully-expanded context-dependent phone model networks for the North American Business News (NAB) task are about twice the size of the corresponding word-level n-gram language model, and so can be used directly without any dynamic expansion in a Viterbi decoder. Runtime savings arise both from the increased determinacy of the model network achieved by our determinization algorithm [8] and from eliminating the run-time overhead of dynamic context-dependent expansion.

2. MODELS

The various levels of recognition modeling are implemented in our system as *weighted finite-state transducers* [1, 2, 4], which are finite-state networks in which each arc is labeled with an input symbol, an output symbol and a negative log probability. Optionally, the input (output) symbol on an arc may be the null symbol ϵ, indicating that the arc does not consume input (produce output). A path in a transducer pairs the concatenation of the input labels on its arcs with the concatenation of the corresponding output labels, assigning the pair the sum of the arc weights.

The transducer representation of models provides a natural algorithm, *composition*, for combining multiple levels of modeling. The composition of two weighed transducers S and T is a transducer $S \circ T$ that assigns the weight w to the mapping from symbol sequence x to sequence z just in case there is some symbol sequence y such that S maps x to y with weight u, T maps y to z with weight v, and $w = u + v$. The states of $S \circ T$ are pairs of a state of S and a state of T, and the arcs are built from pairs of arcs from S and T with paired origin and destination states such that the output of the S arc matches the input of the T arc (null transition labels need to be handled specially) [9, 13]. The transducers in our application are:

Context-dependency transducer C: Maps sequences of names of context-dependent phone models (HMMs) to the corresponding phone sequences. The topology of this transducer is determined by the kind of context dependency used in modeling (e.g. triphonic, pentaphonic, tree-based). For explanatory convenience, the examples and some of the discussion will use the *inverse* C^{-1} of C, which maps phone sequences to HMM name sequences. For example, the transducer shown in Figure 1, C^{-1}, encodes triphonic context dependency for two hypothetical phones x and y. It does not represent a simple substitution, since it describes the mapping from context-independent phones to context-dependent HMMs, denoted here by *phone / left context_right context*. Each state (a, b) encodes the information that the previous phone was a

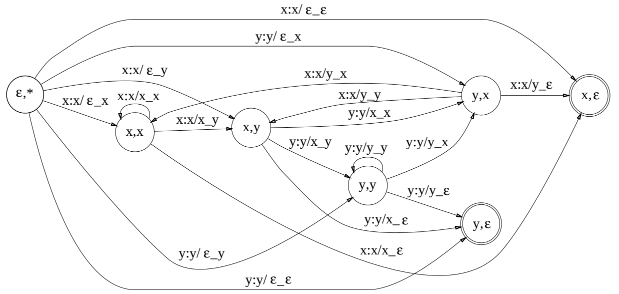

Figure 1: Non-deterministic context-dependency model.

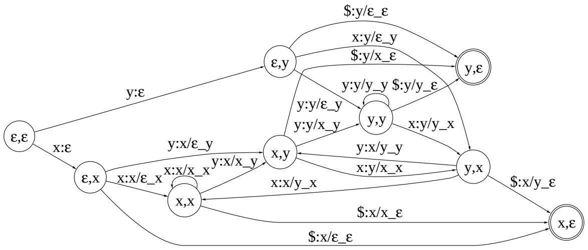

Figure 2: Deterministic context-dependency model.

and the next phone is b; ϵ represents the start or end of a phone sequence and $*$ an unspecified next phone. For instance, it is easy to see that the phone sequence xyx is mapped by the transducer to x/ϵ_y y/x_x x/y_ϵ via the unique state sequence $(\epsilon,*)(x,y)(y,x)(x,\epsilon)$.

Dictionary transducer L: Represents word pronunciations, mapping phone sequences to their possible segmentations into word sequences according to a (possibly multiple) pronunciation dictionary.

Language model G: Represents the probabilities of word sequences, mapping a sequence to itself with a weight corresponding to the language model probability of the word sequence. In general, any finite-state language model could be used, although in practice we use n-gram models, $n = 2, 3$.

Thus each path in the composition $C \circ L \circ G$ pairs a sequence of HMM names with a word sequence, assigning it a weight corresponding to the likelihood that the word sequence is pronounced as specified by the HMM sequence. The composition $C \circ L \circ G$ can thus serve as the modeling network for a standard (e.g. Viterbi) decoder in the usual way.

A distinctive feature of this approach is that context dependency constraints are represented by a transducer C rather than being embedded in the decoder, thus allowing experiments with alternative types of context dependency and alternative ways of combining it with the other models that take advantage of general optimization techniques for weighted automata and do not require any (possibly difficult) changes to the decoder.

3. ALGORITHM

Building the fully-expanded $C \circ L \circ G$ network uses several novel algorithms: efficient transducer composition [9], weighted transducer determinization [7, 8] and ϵ-removal for weighted automata.

Weighted transducer determinization ensures that distinct arcs leaving a state have distinct input labels. Clearly, a necessary condition for transducer determinization is that the initial transducer maps each input sequence to at most one output sequence. But this is not sufficient: the mapping must be *sequential* [1, 7]. These conditions may be somewhat relaxed to mappings with bounded ambiguity (or p-*subsequential* [7]). The purpose of applying determinization to the model network is to decrease the number of alternative arcs that need to be considered during decoding. In many cases, the size of the model is also reduced, because redundant paths are eliminated. Previous work in network optimization [10, 11, 12] has used tree-based constructions that can be seen as limited cases of determinization. General determinization has the following advantages over those approaches: networks need not be constructed as trees, a wider range of networks can be optimized, and the results are in general more compact than trees.

Informally, if the original transducer maps input uv to x with weight c and input uw to y with weight d, then the determinized transducer will admit a unique way of reading u from the initial state. The output sequence associated to u will be the longest common prefix of x and y and the corresponding weight will be $\min\{c, d\}$.

As a first application of determinization, we observe that the natural context-dependency transducer C^{-1} of Figure 1 is not deterministic: a state such as (x, x), for instance, has three outgoing arcs with input label x. However, transducer determinization readily converts it to the deterministic version shown in Figure 2 ($ is a new end-of-utterance symbol used to make the result sequential). Because of this determinization, the inverse C of C^{-1} has a single arc for each output phone leaving each state, which is essential in building a small and efficient $C \circ L \circ G$.

The determinization of $L \circ G$ is the most demanding task in our network optimization method. First of all, neither L nor G is unambiguous. L may map a given phone string to several alternative words because of homophones. G may also have several paths for a given word sequence, for instance when a variable length or backoff language model is interpreted as a finite-state network allowing all the alternative paths corresponding to different context sequences [14]. In both cases, we disambiguate the models by labeling the alternatives with auxiliary symbols (possibly on new arcs), yielding two new transducers L' and G' whose composition $L' \circ G'$ can be determinized. The resulting deterministic transducer P' maps phone strings with interspersed auxiliary symbols to word sequences. The auxiliary labels in P' are now replaced by ϵ and the weighted ϵ-removal algorithm is applied to yield a transducer P. The final fully-expanded model is then $C \circ P$. This transducer is not in general deterministic because the transformation from P' to P can create nondeterminism, but most of the nondeterminism arising from shared phone sequences in the pronunciations of different word sequences will have been eliminated.

In summary, the compilation of the fully-expanded network has the following steps:

1. Determinize the inverse of the context-dependency transducer and invert the result to produce C.
2. Disambiguate L into L' and G into G' by introducing auxiliary labels and transitions.
3. Perform the composition $L' \circ G'$.
4. Determinize $L' \circ G'$ to yield P'.
5. Replace the auxiliary labels in P' by ϵ and remove ϵ-arcs to yield P.
6. Perform the composition $C \circ P$.

4. RESULTS

We used the approach outlined in the previous section to create fully-expanded models for a variety of large-vocabulary recognition tasks, and tested the models in a simple general-purpose one-pass Viterbi decoder. The decoder makes no special provision for context-dependent models, since context-dependency constraints are represented in the transducer C and merged by composition into the overall expanded network. We give the sizes of the individual models and of the intermediate and fully-expanded networks for the North American Business News (NAB) task using bigram and trigram language models and vocabulary sizes that range from 10,000 to 160,000 words, as well as real-time recognition results.

The same context-dependency transducer C is used in all the experiments. The transducer, which has 1523 states and 80,719 arcs, represents triphonic contexts clustered by decision-tree methods that take into account cross-word dependencies [17]. As explained earlier, the input label of each arc in this transducer names an HMM, while the output label names a phone. There are 25,919 distinct HMMs and 5520 distinct HMM states, each associated to a four-gaussian mixture model.

Table 1 lists the lexicon transducer sizes and out-of-vocabulary rates for several vocabulary sizes. For a vocabulary size V, the V most frequent words in the NAB 1994 text corpus were used. [1] The pronunciations for these words were obtained from the AT&T text-to-speech system, and then encoded as the optimized finite-state transducer L.

Table 1: Size of lexicon transducers

Vocab. size	States	Arcs	OOV rate (%)
10000	19146	39976	5.6
20000	37254	78898	2.9
40000	71769	154076	1.4
160000	271356	594145	0.4

Table 2 shows the sizes and test-set perplexities (excluding unknown words) of the various language models used. These were built using Katz's backoff method with frequency cutoffs of 2 for bigrams and 4 for trigrams [3], then *shrunk* with an epsilon of 10 using the method of Seymore and Rosenfeld [16], and finally encoded into (non-deterministic) weighted automata G [14].

Table 3 lists the sizes of the transducers created by composing lexicon transducers with their corresponding language models and determinizing the result, as described in Section 3.

Finally, Table 4 lists the sizes for the transducers created by composing the context-dependency transducer with each of the transducers in Table 3. The resulting transducers represent the fully-expanded networks that are searched during decoding.

We can thus see that the number of arcs in the fully-expanded network is only about 2.1 times that of the language model for bigrams and 2.5 times for trigrams, and so is quite practical for real-time recognition. Moreover, the fully-expanded context-dependent networks in Table 4 are only about 2.5% larger than the corresponding context-independent networks in Table 3. Thus, contrary to conventional wisdom, context-dependency, even with cross-word contexts, does not significantly expand a context-independent phone network if the context-dependency is suitably applied as in our framework.

Table 2: Size and perplexity of language models

Vocab. size	N-gram order	States	Arcs	Perp.
10000	2	10004	1960990	174
20000	2	20004	2591547	194
40000	2	40004	3121446	212
160000	2	160004	3818659	230
10000	3	1861458	7002522	113
40000	3	2771167	9195312	134

Table 3: Size of lexicons composed with language models and determinized

Vocab. size	N-gram order	States	Arcs
10000	2	1381669	4177688
20000	2	1858768	5538887
40000	2	2282180	6681514
160000	2	3050565	8232983
10000	3	7853810	17343182
40000	3	11084228	23474251

Figure 3 shows 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 bigram models above on the DARPA Fall '95 Hub 3 evaluation test set (contrast C0). Figure 4 shows recognition results for trigram models in comparison with results for bigrams of the same vocabulary size. The best word accuracy shown here is 81.2%, while our best off-line system performed at 90.5% word accuracy in the Fall '95 evaluation [6]. The better accuracy of our multipass, non-real-time system can be attributed to more accurate and larger (but slower) acoustic models, gender-dependent models, speaker adaptation, multiple-pronunciation networks, wider search beams, and a 5-gram language model.

In general, we see that larger vocabulary size and n-gram or-

Table 4: Size of fully-expanded context-dependent networks

Vocab. size	N-gram order	States	Arcs
10000	2	1412769	4278821
20000	2	1911112	5679686
40000	2	2352944	6849884
160000	2	3135226	8431949
10000	3	8063802	17799882
40000	3	11353592	24018777

[1] The vocabulary was automatically pre-filtered to remove corpus tokens that were deemed implausible words, for instance those that contained no alphabetic characters.

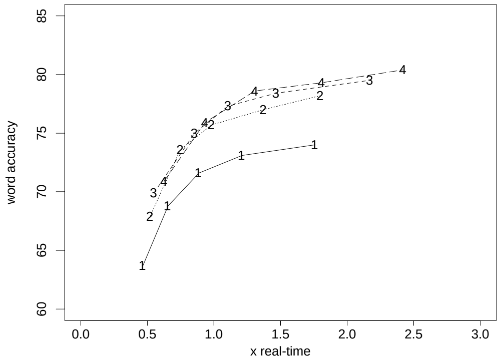

Figure 3: Bigram recognition results for vocabularies of (1) 10,000 words, (2) 20,000 words, (3) 40,000 words, and (4) 160,000 words.

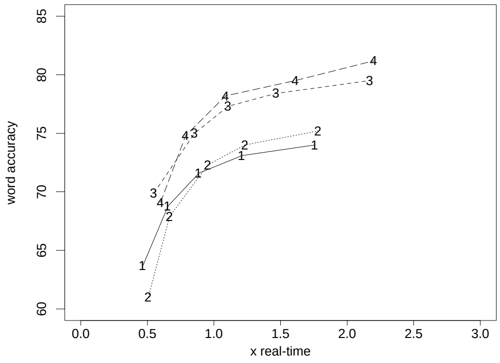

Figure 4: Recognition results for the (1) 10,000 word bigram, (2) 10,000 word trigram, (3) 40,000 word bigram, and (4) 40,000 word trigram.

der give better real-time performance. It is comforting that improved modeling not only gives improved accuracy but also improved speed with our optimized networks. Further, it suggests that adding a rescoring pass just to apply a stronger language model is suboptimal for real-time performance, since we get the best performance by using our strongest language model in our single pass.

5. CONCLUSION

We showed that our approach based on weighted automata provides a new method for creating optimized fully-expanded recognition networks for large-vocabulary recognition with context-dependent phone models. In earlier work, we had shown how the same approach could be used with on-demand model expansion. We thus have a single framework in which both fully-expanded and on-demand models can be built and used efficiently in a simple decoder that stays unchanged even if the context-dependency constraints or network combination method change.

6. REFERENCES

[1] J. Berstel. *Transductions and Context-Free Languages*. Teubner Studienbucher, Stuttgart, Germany, 1979.

[2] S. Eilenberg. *Automata, Languages, and Machines*, volume A. Academic Press, San Diego, California, 1974.

[3] S. Katz. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Trans. of ASSP*, 35(3):400–401, 1987.

[4] W. Kuich and A. Salomaa. *Semirings, Automata, Languages*. Number 5 in EATCS Monographs on Theoretical Computer Science. Springer-Verlag, Berlin, Germany, 1986.

[5] K.-F. Lee. Context dependent phonetic hidden Markov models for continuous speech recognition. *IEEE Trans. ASSP*, 38(4):599–609, Apr. 1990.

[6] A. Ljolje, M. Riley, and D. Hindle. Recent improvements in the AT&T 60,000 word speech-to-text system. In *ARPA Speech and Natural Language Workshop*, Harriman, NY., 1996. Distributed by Morgan Kaufmann, San Francisco.

[7] M. Mohri. On some applications of finite-state automata theory to natural language processing. *Journal of Natural Language Engineering*, 2:1–20, 1996.

[8] M. Mohri. Finite-state transducers in language and speech processing. *Computational Linguistics*, 23, 1997.

[9] M. Mohri, F. Pereira, and M. Riley. Weighted automata in text and speech processing. In *ECAI-96 Workshop*, Budapest, Hungary, 1996.

[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*. Morgan Kaufmann, 1994.

[11] S. Ortmanns, H. Ney, and A. Eiden. Language-model look-ahead for large vocabulary speech recognition. In *ICSLP'96*, pages 2095–2098. University of Delaware and Alfred I.duPont Institute, 1996.

[12] S. Ortmanns, H. Ney, F. Seide, and I. Lindam. A comparison of time conditioned and word conditioned search techniques for large vocabulary speech recognition. In *ICSLP'96*, pages 2091–2094. University of Delaware and Alfred I.duPont Institute, 1996.

[13] F. Pereira and M. Riley. Speech recognition by composition of weighted finite automata. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*, pages 431–453. MIT Press, Cambridge, Massachusetts, 1997.

[14] G. Riccardi, E. Bocchieri, and R. Pieraccini. Non-deterministic stochastic language models for speech recognition. In *Proc. ICASSP*, volume 1, pages 237–240. IEEE, 1995.

[15] M. Riley, F. Pereira, and M. Mohri. Transducer composition for context-dependent network expansion. In *Eurospeech '97*, Rhodes, Greece, 1997.

[16] K. Seymore and R. Rosenfeld. Scalable backoff language models. In *Proceedings of ICSLP*, Philadelphia, Pennsylvania, 1996.

[17] S. Young, J. Odell, and P. Woodland. Tree-based state-tying for high accuracy acoustic modelling. In *ARPA Human Language Technology Workshop*, 1994. Distributed by Morgan Kaufmann, San Francisco.