# Fast On-The-Fly Composition for Weighted Finite-State Transducers in 1.8 Million-Word Vocabulary Continuous Speech Recognition

*Takaaki Hori, Chiori Hori*[1]*, and Yasuhiro Minami*

NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan
{hori,minami}@cslab.kecl.ntt.co.jp, chiori@cs.cmu.edu

## Abstract

This paper proposes a new on-the-fly composition algorithm for Weighted Finite-State Transducers (WFSTs) in large-vocabulary continuous-speech recognition. In general on-the-fly composition, two transducers are composed during decoding, and a Viterbi search is performed based on the composed search space. In this new method, a Viterbi search is performed based on the first of two transducers. The second transducer is only used to rescore the hypotheses generated during the search. Since this rescoring is very efficient, the total amount of computation in the new method is almost the same as when using only the first transducer. In a 30k-word vocabulary spontaneous lecture speech transcription task, our proposed method significantly outperformed the general on-the-fly composition method. Furthermore the speed of our method was slightly faster than that of decoding with a single fully composed and optimized WFST, where our method consumed only 20% of the memory usage required for decoding with the single WFST. Finally, we have achieved one-pass real-time speech recognition in an extremely large vocabulary of 1.8 million words.

## 1. Introduction

In recent years, large-vocabulary continuous-speech recognition (LVCSR) systems have been incorporated into various speech applications, such as dictation systems, speech dialogue systems, broadcast news captioning systems and so on. The decoding process of LVCSR finds a sequence of words that best matches an input signal from among many hypotheses. Although the above applications work with current technologies, more efficient search algorithms are still needed for very-large-vocabulary and spontaneous-speech transcription tasks. In such tasks, it is more difficult to distinguish a more likely hypothesis because many ambiguous hypotheses are generated during the search.

Recently, the Weighted Finite-State Transducer (WFST) approach has become a promising alternative formulation to traditional decoding approaches, because it offers a unified framework representing various knowledge sources, and it produces a full search network optimized up to the HMM states [1]. The optimization step minimizes search space and accelerates decoding.

However, in the case of using a very large vocabulary lexicon, a detailed language model, or complicated transduction model, an enormous huge transducer is usually generated by composing all the components. Accordingly, both the amount of computation and the memory usage in decoding increase even if the WFST is optimized. Especially for memory usage, the problem is serious. If transducers are composed, it is possible that the size of the resulting transducer will be equal to the product of those transducers. Thus, the size of such a composite transducer easily exceeds the limitations of standard personal computers.

On-the-fly composition is a practical alternative to avoid such a huge memory requirement [2][3]. In on-the-fly composition, the set of WFSTs are separated into two or more groups, and in each group, one WFST is composed and optimized. Composition between the groups is performed during decoding if necessary. In [2] and [3], a lot of memory was saved by using on-the-fly composition, but search efficiency was decreased due to composition overhead.

In this paper, we propose a new on-the-fly composition algorithm to achieve fast and memory-efficient decoding. In the new method, a Viterbi search is performed based on the first transducer of the two groups, whereas the second transducer is only used to rescore the hypotheses generated during the search. Since computation of this rescoring is minimal, the total computation amount of the new method is almost the same as when using only the first transducer. We conducted experiments in a spontaneous lecture speech transcription task and in an extremely large-vocabulary recognition task in a spoken interactive open domain question-answering system [7]. The results show that the proposed method outperforms traditional methods.

## 2. Weighted Finite-State Transducers in Speech Recognition

Recently, the WFST approach has been widely used for speech recognition. WFSTs are finite state networks associating input and output symbols on each arc that can be weighted with a log probability value.

Speech recognition is a transduction process from speech input to the corresponding word sequence. The process can be represented as a cascade of several transductions, each of which can be written in WFST form. Those WFSTs can then be combined by using a composition operator, leading to the integration of the underlying knowledge sources into a single input-output relation. An integrated WFST for speech recognition can be composed as

$$R = H \circ C \circ L \circ G, \tag{1}$$

where $H$, $C$, $L$, and $G$ are WFSTs for a state network of triphone HMMs, a set of connection rules for triphones, a pronunciation lexicon, and a trigram language model, respectively; "$\circ$" represents the composition operator. As a result, decoding is a one-pass search problem for a single huge network $R$ including cross-word triphones and a trigram language model. Once the network is further optimized by proceeding to weighted determinization and minimization, search efficiency dramatically increases.

[1] Currently the author works for Language Technologies Institute, School of Computer Science, Carnegie Mellon University.

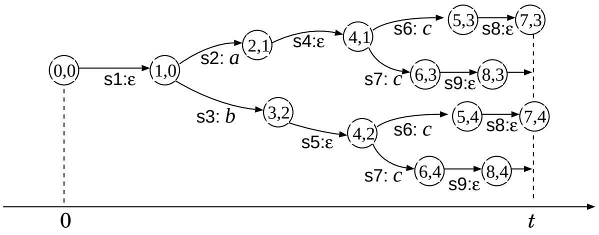

Figure 1: Hypotheses in standard on-the-fly composition

## 3. On-The-Fly Composition

When a single WFST is composed of all knowledge sources for LVCSR, the number of states and transitions often becomes so large that an enormous amount of memory is required during decoding. To avoid this problem, on-the-fly composition is available. Generally in on-the-fly composition two WFSTs are prepared, which are composed during decoding.

In [3], the WFSTs are divided into two groups:

$$(H \circ C \circ L \circ G_{uni}) \circ G_{tri/uni}. \qquad (2)$$

The composite transducer for the first group $(H \circ C \circ L \circ G_{uni})$ translates a HMM-state sequence into the corresponding word sequence, while the other transducer $G_{tri/uni}$ assigns trigram probabilities to the word sequence, where $G_{uni}$ is a unigram model and $G_{tri/uni}$ is the trigram model adjusted by dividing each trigram probability by the unigram probability of $G_{uni}$. As stated in [2], pushing the weights over groups is a kind of "look-ahead" technique for improving search efficiency.

Figure 1 shows a decoding process with standard on-the-fly composition of two transducers, HMM-state-to-word transducer $(H \circ C \circ L \circ G_{uni})$ and language model transducer $G_{tri/uni}$, which are illustrated in Figs. 2 and 3, respectively.

In Fig. 1, the nodes and arcs indicate different hypotheses along the time axis. A pair of numbers in each node means that this node is composed of two states, one of which is from the first transducer and the other is from the second transducer. For example, node (2,1) means that this node is composed of state 2 in Fig. 2 and state 1 in Fig. 3. Each arc is also composed of two transitions from the two transducers. The lefthand side of ":" indicates an input symbol (an index of a HMM-state), and the righthand side of ":" indicates an output symbol (word); "$\epsilon$" means that nothing is output.

Although the number of hypotheses increases according to the combined states as shown in Fig. 1, memory usage is much saved since the combined states are generated only when those states are necessary during decoding. However, the actual search space is usually larger than that of the full-composition method since it is difficult to optimize the search space before decoding. Furthermore, the overhead of on-the-fly composition also increases the amount of computation required for decoding.

## 4. Fast On-The-Fly Composition

We propose a new on-the-fly composition algorithm for fast and memory-efficient speech recognition. The concept of the proposed method is as follows.

Suppose there are two transducers $A$ and $B$ that can be composed. In general on-the-fly composition, given an input symbol sequence $X$, the decoder finds $\hat{Z}$ such that

$$W(X \to \hat{Z}) = \max_{Y,Z} \left\{ W_A(X \to Y) + W_B(Y \to Z) \right\}, \qquad (3)$$

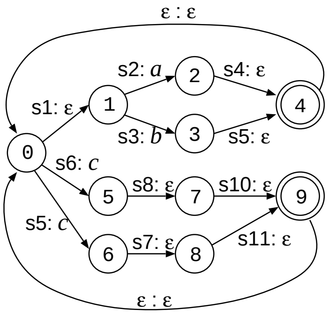

Figure 2: HMM-state-to-word transducer

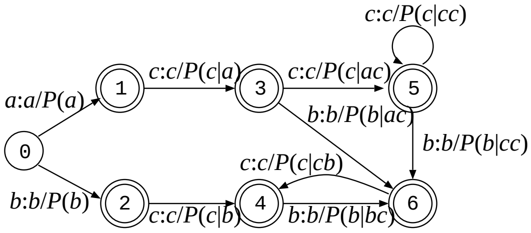

Figure 3: Language model transducer

where $W_A(X \to Y)$ is the accumulated score when the transducer $A$ translates the symbol sequence $X$ to $Y$; $W_B(Y \to Z)$ is also the score of transduction $Y \to Z$ by the transducer $B$. The recognition result is the output symbol sequence $\hat{Z}$ that derives $W(X \to \hat{Z})$.

Equation (3) can be rewritten as

$$W(X \to \hat{Y}) = \max_{Y} \{ W_A(X \to Y) + \max_{Z} W_B(Y \to Z) \}. \qquad (4)$$

This equation means that the algorithm for finding $\hat{Y}$ can be applied to obtain $W(X \to \hat{Z})$ which is equal to $W(X \to \hat{Y})$, where $\max_Z W_B(Y \to Z)$ can be assumed as the compensation score.

In the new method, a Viterbi search is performed based on the first transducer $A$ but not based on the composite transducer $A \circ B$. In frame-synchronous processing in the Viterbi search, hypotheses are generated by $A$, each of which represents an individual state transition process in $A$. If a new hypothesis $h$ is generated by adding a new transition $e$ to an existing hypothesis, $h$ is rescored by

$$\max_f W_B(o[h] \to o[f]),$$

using the second transducer $B$ only when the transition $e$ has a non-epsilon output symbol, where $f$ indicates a hypothesis generated by $B$ accepting $o[h]$ which means the output symbol sequence of $h$. $o[f]$ means the output symbol sequences of $f$ as well.

By associating each hypothesis $h$ with a list of hypotheses $g[h]$ produced by $B$, the rescoring process can be efficiently performed. Here, $g[h]$ means the set of hypotheses that are generated by $B$ when $o[h]$ is given as an input symbol sequence for $B$. We call the hypotheses in $g[h]$ produced by $B$ "co-hypotheses" to distinguish them from the hypotheses produced by $A$ in the Viterbi search.

Suppose a new hypothesis $h'$ is generated by adding a transition $e$ from the state $n[h]$ that $h$ has reached in the transducer $A$. The score of $h'$ derived with $A$ is

$$\alpha_A(h') = \alpha_A(h) + w(e), \qquad (5)$$

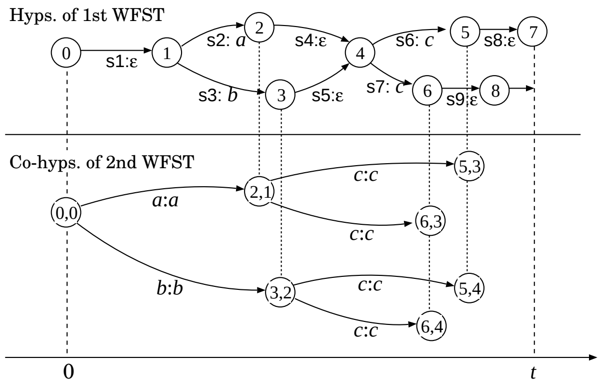

Figure 4: Hypotheses in proposed on-the-fly composition

where $w(e)$ indicates the weight of transition $e$.

If transition $e$ outputs nothing ($e$ has an epsilon output symbol), $g[h]$ does not change, i.e. $g[h'] = g[h]$ since $o[h]$ does not change. On the other hand, if transition $e$ outputs a non-epsilon symbol $y$, a new co-hypothesis $f'$ is generated for each co-hypothesis $f$ in $g[h]$ by adding a transition $r$ which accepts symbol $y$ from the state $n[f]$ that $f$ has reached. The score of $f'$ can be calculated as

$$\alpha_B(f') = \alpha_B(f) + w(r). \tag{6}$$

New co-hypotheses generated by the above procedure are then stored in $g[h']$.

Accordingly, the Viterbi search is performed based on the score:

$$\alpha(h') = \alpha_A(h') + \max_{f' \in g[h']} \alpha_B(f'). \tag{7}$$

During the search, when different hypotheses meet at the same state in $A$, only the best hypothesis survives and then their co-hypothesis lists are merged. If there are different co-hypotheses which have reached at the same state in $B$, only the best co-hypothesis among them is retained in the merged list.

At the end of the utterance, the best complete hypothesis can be derived as

$$\hat{h} = \operatorname*{argmax}_{h:n[h] \in F_A} \alpha(h), \tag{8}$$

and the best complete co-hypothesis can be derived as

$$\hat{f} = \operatorname*{argmax}_{f \in g[\hat{h}]:n[f] \in F_B} \alpha_B(f), \tag{9}$$

where $F_A$ and $F_B$ indicate sets of the final states of transducers $A$ and $B$, respectively. Accordingly, the recognition result is $o[\hat{f}]$, i.e. the output symbol sequence of $\hat{f}$.

Figure 4 shows a decoding process in the proposed on-the-fly composition when the decoder uses the transducers in Figs. 2 and 3. The upper half of Fig. 4 represents a set of hypotheses generated by the first transducer in Fig. 2. Compared to the case of standard on-the-fly composition in Fig. 1, the number of hypotheses is much smaller. As shown in the lower half of Fig. 4, each hypothesis is linked to a set of co-hypotheses that is based on the second transducer, and it is rescored by those co-hypotheses. Since minimal computation is required to update the list of co-hypotheses, the total amount of computation is almost the same as when decoding with only the first transducer.

In the proposed method, rescoring with co-hypotheses is effective for pruning hypotheses during the search. By modifying the score of each hypothesis with the corresponding co-hypotheses, each hypothesis can be accurately evaluated using all knowledge sources. Hence, more promising hypotheses can be kept in a beam search compared to multi-pass search strategies in which, for example, only the first transducer is used in the first pass.

However, the proposed method does not necessarily ensure that the best hypothesis is found because different co-hypotheses share the same time alignment. In the proposed method, it is assumed that the time alignment of the transition (4,1), (6,3), (8,3) in Fig. 1 is always equal to that of (4,2), (6,4), (8,4) in the same figure. Of course, this assumption is not always correct because it is possible that the time at node (4,1) is not equal to the time at node (4,2).

As mentioned in [4], however, triphones yield a good assumption. When we use triphones, transitions to a state come from states associated with a unique preceding phone. As a result, during decoding, time alignment is retained depending on the preceding phone. Although the preceding phone is not necessarily unique, since the WFST is actually optimized up to the shared HMM-states, we can say that *phone-pair approximation* is roughly used. The phone-pair approximation assumes that the best starting time for a phone only depends on the preceding phone rather than on the entire preceding phone sequence. If this assumption is satisfied, our method ensures that the best hypothesis is found.

## 5. Evaluation with CSJ Task

We evaluated our on-the-fly composition method in a 30k-word spontaneous speech transcription task. The task is based on a corpus of spontaneous Japanese (CSJ) [5], mostly comprising monologues such as lectures, presentations, and news commentaries.

The evaluation data were limited to presentations in academic fields. The speeches were digitized with 16-kHz sampling and 16-bit quantization. Feature vectors had 25 elements consisting of 12 MFCCs, their delta components and a delta log energy. Tied-state triphone HMMs with 3,000 states and 16 Gaussians per state were made by using 787 presentations in the corpus uttered by male speakers (approximately 187 hours). A trigram language model was estimated using manually transcribed text data of 2,592 presentations. Benchmark test 1 is used for evaluation, which consists of ten academic talks presented by male speakers. The test-set perplexity is 78.7, and the out-of-vocabulary rate is 2.2%.

We used a speech recognizer *SOLON* [6] developed at NTT Communication Science Laboratories, which performs a one-pass Viterbi search based on a single WFST or two WFSTs that can be composed. A standard PC (with a Xeon 3.0-GHz processor) was used to measure the speed of the decoder.

Figure 5 shows the relationship between word accuracy (WACC) and decoding time in each method when changing the beam width parameter of the decoder. The decoding time is represented by a real-time factor (RTF) that indicates the ratio of decoding time to utterance time. Our proposed method outperformed the full composition method. In this case, our method required only 20% memory usage of the full composition method. In addition, while our method achieves the same accuracy as standard on-the-fly composition, it is 1.5 to 2 times faster.

## 6. Evaluation with ODQA Task

We conducted additional experiments on the task for a spoken interactive open-domain question-answering (ODQA) system developed at NTT Communication Science Labs [7]. In this task, a user asks the system a question that domain is not restricted, after which the system finds the answer from a large corpus of news texts covering the last 12 years. Since the system cannot know what the user

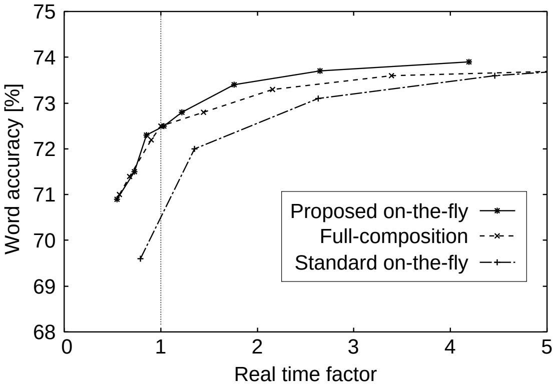

Figure 5: WACC vs. RTF in CSJ task

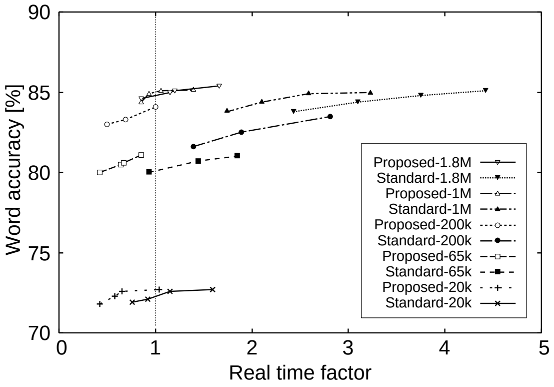

Figure 6: WACC vs. RTF in ODQA task

will ask in advance, the speech recognizer has to cover an extensive vocabulary.

We developed five sets of lexicon and language models that corresponded to 20K, 65K, 200K, 1M, and 1.8M vocabularies. These models were estimated using newspaper articles of the last 12 years and about 14,000 interrogative sentences.

Tied-state triphone HMMs with 3,000 states and 16 Gaussians per state were trained by using read speech data uttered by about 150 female speakers (approximately 50 hours). The evaluation data consists of 2,000 questions uttered by a female speaker who does not belong to the training data. We show the complexity of this task in Table 1.

We used a computer (with an Opteron 246 2-GHz processor and a 16-Gbyte memory) to compile transducers and to measure the speed of the decoder. We tried to build the full-composite transducers, but it was no longer possible. Thus, in the experiment, we just compared standard on-the-fly composition method with our proposed method.

Figure 6 shows the relationship between word accuracy and decoding time in each method. These results reveal that the proposed method is 2 to 3 times faster than the standard method. Since the margin increases as the vocabulary size expands, our method especially recognizes a very large amount of vocabulary speech. Finally, we have achieved real-time speech recognition of a vocabulary of 1.8 million words.

## 7. Conclusions

In this paper, we have proposed a new on-the-fly composition algorithm for Weighted Finite-State Transducers (WFSTs) in LVCSR. In a 30k-word vocabulary spontaneous lecture speech transcription task, our proposed method outperformed not only general on-the-fly composition, but also decoding based on a single WFST that is fully composed and optimized. In that task, our method consumed only 20% of the memory usage required for decoding with a fully compiled WFST. In addition, we have also achieved real-time speech recognition in an extremely large vocabulary of 1.8 million words.

Table 1: Out of vocabulary rate and test-set perplexity in ODQA task

| Vocabulary size | 20K | 65K | 200K | 1M | 1.8M |
|---|---|---|---|---|---|
| OOV rate [%] | 8.0 | 3.4 | 1.9 | 0.8 | 0.6 |
| Perplexity | 100.1 | 128.0 | 150.4 | 169.4 | 177.1 |

Since this new method is a general algorithm for on-the-fly composition of WFSTs, it can be applied to not only speech recognition, but also other processings. In the future, we would like to apply this technique to speech-input language processing such as speech summarization [8], speech translation and so on.

## 8. Acknowledgement

We thank the Japanese Science and Technology Agency Priority Program, “Spontaneous Speech: Corpus and Processing Technology,” for providing speech data and transcriptions. Furthermore we thank MIT’s SLS group for providing the tools composing and optimizing WFSTs.

## 9. References

[1] M. Mohri, F. Pereira, M. Riley, “Weighted finite-state transducers in speech recognition,” Proc. of ASR2000, pp. 97–106, 2000.

[2] H. J. G. A. Dolfing, I. L. Hetherington, “Incremental language models for speech recognition using finite-state transducers,” Proc. of ASRU2001, 2001.

[3] D. Willett, S. Katagiri, “Recent advances in efficient decoding combining on-line transducer composition and smoothed language model incorporation,” Proc. of ICASSP2002, Vol. I, pp. 713–716, 2002.

[4] A. Ljolje, F. Pereira, M. Riley, “Efficient general lattice generation and rescoring,” Proc. of Eurospeech 1999, pp. 1251–1254. 1999.

[5] T. Kawahara, H. Nanjo, T. Shinozaki, S. Furui, “Benchmark test for speech recognition using the corpus of spontaneous Japanese,” Proc. of SSPR2003, pp. 135–138, 2003.

[6] T. Hori, “NTT Speech recognizer with OutLook On the Next generation: SOLON,” Proc. of CSA2004, 2004.

[7] C. Hori, T. Hori, H. Isozaki, E Maeda, S. Katagiri, S. Furui, “Deriving disambiguous queries in a spoken Interactive ODQA system,” Proc. of ICASSP2003, Vol. I, pp. 624–627, 2003.

[8] T. Hori, C. Hori, Y. Minami, “Speech summarization using weighted finite-state transducers,” Proc. of Eurospeech 2003, pp. 2817–2820, 2003.