2004－SLP－54 (43)
2004／12／22

高次の音素環境依存モデルを統合した
重み付き有限状態トランスデューサの効率的生成法

シュスター・マイク[†]　堀貴明[†]

[†] 日本電信電話（株）ＮＴＴコミュニケーション科学基礎研究所　〒619–0237 京都府相楽郡精華町光台 2–4
E-mail: †{schuster,hori}@cslab.kecl.ntt.co.jp

あらまし　本稿では，トライフォン (triphone) を越える高次の音素環境依存モデルを統合した重み付き有限状態トランスデューサ (Weighted Finite State Transducer : WFST) を効率的に生成するアルゴリズムについて述べる．音声認識用の WFST を構築する従来の手法では，高次の音素環境依存モデルを扱う際にいくつかの問題に直面する．そして，場合によっては計算量やメモリの制約から構築そのものが不可能になることがある．本稿では，まず，従来の構築方法の非効率性について議論した後で，音声認識用 WFST の構築において必要とされる，HMM 状態列を音素列へ変換する WFST を音素決定木から直接生成する効率的なアルゴリズムを提案する．そして，そのアルゴリズムが非常に高速かつ省メモリで動作し，かつ，最終的に構築される WFST のサイズをも小さくすることを示す．提案手法により構築された WFST を，日本語話し言葉コーパスを用いて，サイズ，認識精度，認識速度の観点から評価した．そして，高次の音素環境依存モデルを組み込んだ単語内・単語間の音素環境を考慮するワンパス時間同期探索が容易に実現され，かつ，それが triphone の場合に比べてわずかなオーバヘッドで動作することを示す．最後に，単語内・単語間の quinphone を適用した実時間音声認識が 125MB のメモリかつ 9%のサーチエラーで動作することを確認した．

キーワード　音声認識, 重み付き有限状態トランスデューサ, 音素環境依存モデル

Efficient Generation of high-order context-dependent
Weighted Finite State Transducers for Speech Recognition

Mike SCHUSTER[†] and Takaaki HORI[†]

[†] Nippon Telegraph and Telephone Corporation, NTT Communication Science Laboratories
2–4 Hikari-dai, Seika-cho, Soraku-gun, Kyoto-fu, 619–0237 Japan
E-mail: †{schuster,hori}@cslab.kecl.ntt.co.jp

Abstract　This paper describes an algorithm for efficient building of Weighted Finite State Transducers for speech recognition when high-order context-dependent models of order $K > 3$ (triphones) with tied states are used. After discussing some inefficiencies of the standard compilation method which make the use of high-order context-dependent models cumbersome and sometimes even impossible because of memory constraints, we show how an algorithm to build a part of the needed composed transducers directly from the decision trees in combination with an improved compilation process can lead to much faster, simpler and more memory-efficient compilation. In our case it also resulted in substantially smaller final networks. With the described algorithm it is simple to use high-order full cross-word models with little overhead directly within a one-pass time-synchronous search, which we test comparing resulting final network sizes, recognition rates and speed on a large, spontaneous Japanese speech database. Using the proposed algorithm it is possible to do real-time recognition using full cross-word quinphones with a large acoustic model in about 125MB of memory at about 9% search error.

Key words　Speech recognition, search, weighted finite state transducers

1. Introduction

In almost all speech recognition systems context-dependent (CD) acoustic models are used since their use reduces assumptions and improves recognition rates over using context-independent models. In most cases these are triphones (order $K = 3$) observing a context of +/- 1 phone, but higher order models have been in use as well since it often leads to additional improvements [1] [2]. We observed that especially often occurring words like for example function words and digits usually benefit from higher-order models because these words' high occurrence frequencies in the training data will often result in unique observation state sequences for them when used with decision-tree tied-state acoustic models. This in turn leads to a higher proportion of parameters allocated to these often occurring words and therefore usually to better overall recognition rates.

The main reason for not using models of higher order than triphones are the difficulties encountered during training and decoding. There are some minor practical issues related to training of CD models of order $K > 3$ which are briefly discussed below. The difficult part with respect to high-order CD models is to make proper use of them during decoding within and in-between words (cross-word models). For large scale systems higher-order models are usually used in rescoring passes [1] [2] where the decoding process is relatively simple because of the limited number of hypotheses at any time. Among all the methods proposed for decoding with context-dependent models so far the probably most elegant is the handling of context-dependency within the *Weighted Finite State Transducer* (WFST) framework [3] [4] [5]. The principal advantage when using WFSTs is that the decoding process is completely decoupled from dealing with context-dependency since the CD models are compiled in advance into a network that treats context-independent and context-dependent (cross-word) models of any order exactly in the same way. The difficulty lies in the compilation process itself when higher order models are used.

In section 2. we first discuss some minor practical problems when training CD models of high context order. For usage of these models in a WFST framework we discuss in section 3. the traditional way of handling CD models, highlight some of its disadvantages which can become severe for context-dependent models of order $K = 5$ and greater, and then present a more efficient method which allows to compile WFSTs a lot faster, more memory efficiently, and in our case, into a lot smaller and therefore more efficient final networks.

In section 4. we report on results of experiments with different order models using a large Japanese spontaneous speech database (CSJ lecture speech corpus) [6] and compare resulting network sizes, recognition speeds and recognition accuracies for different setups, which show that high-order models can be used without much overhead right from the start (and not only in a rescoring pass) in a one-pass time-synchronous search.

2. Issues in high-order context-dependent model training

Here we discuss briefly three minor issues when training high-order CD models: 1) the expansion into context-dependent state-graphs, 2) the collection of statistics for phonetic decision trees, and 3) the building of the phonetic decision trees.

- **Expansion into context-dependent state-graph:** When forward-backward training is used the occurring phone graphs during training (containing optional pauses and possibly multiple pronunciations for ML training, and additionally all competing hypotheses in case of discriminative training for the denominator) are in our case expanded on the fly into their corresponding state-graphs of the required context order. We don't expand the silence and noise models.
- **Collection of statistics for phonetic decision trees (PDTs):** When using the A-set of the lecture speech training data of the CSJ corpus ($230h$ of speech) the number of occurring unique contexts for quinphones is about $487k$ compared to only $14k$ for triphones, and for all of these (times the number of used states per phone, in our case three) Gaussian statistics need to be collected and stored, which adds up to an annoyingly large $> 500MB$ in case of quinphones and 39-dimensional features. We collect the PDT statistics using Viterbi alignment which can be done fairly quickly (15 min in our case) since it can be distributed over a large number of machines. In our case summing up the statistics from all machines takes by far the most time which we reduced significantly using a hierarchical summation scheme.
- **Building the phonetic decision trees:** We build one PDT per phone and state position as used in [7]. In our case a roughly 5-fold speed-up of the method described in [7] is achieved by only adding up the "YES" statistics for each proposed split and calculating the "NO" statistics by subtracting the "YES" statistics from the parent statistics. To build all decision trees for quinphones using $1.46M$ Gaussians takes then roughly 15 min.

3. Issues in high-order context-dependent model usage

Decoding with context-dependent models is conceptually simple in a WFST framework since all information is compiled in advance (or on-the-fly) into one large WFST taking state IDs as inputs and producing word IDs as outputs. The difficulty lies in the compilation process itself.

The method described in [3] of building the complete network mapping state observation IDs to words involves building four separate transducers first:

- transducer H mapping (sequences of) state IDs to (sequences of) CD models
- transducer C mapping (sequences of) CD models to (sequences of) phones
- transducer L mapping (sequences of) phones to (sequences of) words, also called the *lexicon*
- transducer G mapping (sequences of) word IDs to (sequences of) word IDs with probabilities, also called the *grammar* or *language model*

The three transducers H, C, and L are extended to their *determinizable* versions $\tilde{H}$, $\tilde{C}$ and $\tilde{L}$ by introduction of auxiliary symbols as explained in [3]. The four transducers are then compiled into a single transducer X as follows:

$$X = fact(\pi_\epsilon(min(det(\tilde{H} \circ det(\tilde{C} \circ det(\tilde{L} \circ G)))))) \qquad (1)$$

with the occurring operations defined as explained in [3]:

- $A \circ B$: weighted composition of A and B
- $det(A)$: weighted determinization of A
- $min(A)$: weighted minimization of A
- $\pi_\epsilon(A)$: replacement of auxiliary symbols by ϵ in A
- $fact(A)$: factorization of A

The method described above needs an FST H mapping state IDs to CD models and another FST C to map CD models to phones. The disadvantage of using two separate FSTs is that usually phonetic decision trees are used to determine clustered state IDs shared among many models where CD model names don't have much meaning anymore - many different CD models will map to the same or similar state ID sequences after optimization of the resulting network, but this is ignored when building the individual FSTs. The size of C grows exponentially with the context order K - for K-phones it has N^{K-1} states and N^K arcs for a phone set of size N. Even when limiting the size of C and H by using only the contexts occurring in the dictionary (for high K these will also be many because of cross-word effects), generation and subsequent processing of C and H is cumbersome for context order $K > 4$ and $N > 40$ and makes it virtually impossible to use CD models with $K > 5$ because of memory constraints during the final optimization stages of Eq.(1) when using a large L and G.

It is much more efficient to generate the determinizable composition of H and C ($H \circ C$) directly from the PDTs as shown below and attach then the necessary auxiliary symbols. This will alter the overall compilation process given in [3] and resulted in our case besides being much faster (total generation time of $H \circ C$ is only a few seconds for triphones to minutes for quinphones) and more memory-efficient also in substantially smaller final FSTs of about 20-25% the original size.

3.1 Direct Construction of Transducer $H \circ C$

Transducer $H \circ C$ mapping sequences of observation state-IDs directly to context-independent phone sequences is directly constructed as follows assuming context-order K and number of basic phones N with a total of S phonetic decision tree leaf nodes altogether. We denote the resulting transducer as $\tilde{HC}$ which includes the auxiliary symbols needed for determinizing its composition with the other transducers L and G in later composition stages.

3.1.1 Step 1: Generation of PDT leaf node bitmaps

For each PDT leaf-node generate a two-dimensional bitmap that describes what phone n is allowed at what context-order position k, such that bitmap $B_s[n][k] = 1$ only when phone n allowed at position k. This can be done efficiently in a recursive procedure going down the phonetic decision tree. Each of the S bitmaps describes the set of all possible K-phone order CD models which belong to this PDT leaf node.

	position				
phone	0	1	2	3	4
a	1	1	0	1	0
a:	0	1	1	0	1
b	1	1	0	1	0
by	0	1	0	1	0
.			.		
.			.		
z	0	1	0	1	0

Tab. 1 Bitmap example for one PDT leaf node for context-order $K = 5$ (quinphones). In this example phone a: is the center phone and for example phone z is only allowed at positions right before and after the center phone.

3.1.2 Step 2: Generation of bitmaps and state ID sequences for each phone

For each phone n generate all possible state ID sequences (usually three states long) and their corresponding bitmaps by *multiplication* (binary AND operation of bitmaps) of all possible combinations of the basic bitmaps for each state position within that phone. If the resulting bitmap has at every position k at least one phone active (at least one '1' per column), the corresponding CD model set has at least one member and therefore is a valid combination. For three states this can be written for phone n as:

$$B_{n,j} = B_{s1} \,\&\, B_{s2} \,\&\, B_{s3} \qquad (2)$$
$$\forall\ s\{i\} \in S(\text{state pos} = i,\ \text{phone} = n)$$

This step is done efficiently by first generating all combinations of the first and second state, then filtering out the invalid combinations, and then combining the result with the third state (and so on if more states are used). If not done this way, the number of combinations grows exponentially with the number of state positions in the phone and can even for only three states become lengthy to calculate since the number of leaf nodes for a specific state position and phone can be up into the hundreds generating possibly millions of combinations, which will mostly be invalid. Given a total of a few thousand states S there will be at most a few hundred to the low thousands (denoted by J_n) valid state sequences per center phone.

We store all valid resulting state ID sequences in a hash table indexed by center phone and counter j per center phone since they will be needed in later steps again.

3. 1. 3 Step 3: Generate within-phone connections

Generate the **within-phone connections** of transducer $\tilde{HC}$ for each resulting valid state ID sequence $B_{n,j}$ as a simple sequence of arcs with state ID as input and ϵ as output, and add optionally auxiliary symbol loops (see [3] for their meaning) at the first state of each sequence.

3. 1. 4 Step 4: Generate between-phone connections

Generate **between-phone connections** of $\tilde{HC}$ by connecting the sequences generated in the previous step with arcs having ϵ as input and the phone of the sequence it is connecting to as output. To find out which sequences connect to other sequences that are in fact valid combinations allowed by the original decision tree, multiply a *binary right shifted* (symbol '>>' in C, C++) version of all bitmaps from step 2 with all non-shifted bitmaps from step 2 – if the resulting bitmap has at least one phone left at all positions but the first one (there cannot be any phone left at the first position because it was masked out by the shifted bitmap), then it is a valid connection. This can be written as:

$$\begin{aligned} B_{n1,j1,n2,j2} &= (B_{n1,j1} >> 1)\ \&\ B_{n2,j2} \qquad (3) \\ &\forall\ n1, n2 \in N \\ &\forall\ j1, j2 \in J_n \end{aligned}$$

3. 1. 5 Step 5: Generate initial/final state connections

Generate **initial/final state connections** by connecting them to the sequences for which left/right context is unknown (initial arcs same as in step 4, final arcs just ϵ/ϵ), and optionally leave some of these out if you want to force recognition to start/stop with only certain phones.

3. 2 New Compilation Process

The overall compilation process Eq.(1) needs to change since there are no separate transducers H and C anymore. It is now:

$$X = fact(\pi_\epsilon(min(det(det(\tilde{HC}) \circ det(\tilde{L} \circ G))))) \qquad (4)$$

As can be seen it involves separate determinization of $\tilde{HC}$, which will be very beneficial when tied-state models are used since many state sequences will share a common beginning, and $\tilde{L} \circ G$ as well as of their composition $det(\tilde{HC}) \circ det(\tilde{L} \circ G)$.

4. Experiments & Results

We used the A-set (male & female) lecture speech subset (186k utterances, 230h) of the CSJ Japanese spontaneous speech database for training and test set 1 (10 lectures, 26515 words, perplexity 78.7 for trigram, out-of-vocabulary rate 2.4%) with a 30k dictionary and trigram for testing [6]. Preprocessing is standard 39-dimensional MFCCs, all states have 16 diagonal Gaussians each. All models were trained from flat start with an identical training procedure which was known to give non-optimal individual results (e.g. PDT statistics were collected from only monophone alignments) but allows fair comparison in a reasonable amount of time. To avoid search errors, all experiments were run with a large beam (250).

model	size(C)	size($det(\tilde{HC})$)
K/S	(#nodes/arcs)	(#nodes/arcs)
2/1k	43/1.8k	2k/22k
2/2k		3.1k/35k
3/1k	1.8k/80k	4.4k/16k
3/2k		8.6k/25k
3/3k		12k/31k
3/4k		14k/35k
3/5k		16k/39k
4/2k	80k/3.4M	12k/36k
4/3k		18k/55k
4/4k		24k/72k
4/5k		31k/92k
5/2k	3.4M/147M	18k/60k
5/3k		31k/99k
5/4k		45k/142k
5/5k		58k/184k

Tab. 2 Sizes of full C and determinized $\tilde{HC}$ without auxiliary symbols for models with context order K and S states for CSJ test set 1 conditions with basic phone set size $N = 43$.

Tab.2 shows the hypothetical sizes of the full transducer C and intermediate determinized transducer $\tilde{HC}$ in our case using the proposed algorithm still containing all possible mappings from state ID sequences to phone sequences allowed by the acoustic model. As can be seen, the exponential explosion of the complexity of C is clearly a problem.

For recognition, **Tab.3** shows that models of similar complexity produce better results the more context is observed although the actual absolute improvement in recognition rate is rather small. Final network size and decoding time increase only moderately for higher context order due to the large amount of sharing of CD models and corresponding state sequences.

In our case, experiments for quart- and quinphones were only possible when using our algorithm to directly generate $\tilde{H}C$ because of the otherwise too large memory and/or time demands of the standard composition algorithm Eq.(1) as outlined above. But even when the new algorithm was used for triphones we observed besides much faster overall compilation a large reduction in final network size (of about 75 – 80%) which showed that our original composition procedure based on Eq.(1) was suboptimal.

As shown in the last three rows of **Tab.2**, real-time decoding with full cross-word quinphones is possible at 9% search error using a reduced beam and on-the-fly composition [8] in less than 125 MB total memory.

model K/S	size(X) (#arcs)	LM weight	error	RTF
2/1k	2.21M	14	26.9%	4.8
2/2k	2.22M	14	26.8%	5.8
3/1k	2.47M	14	24.8%	6.0
3/2k	2.80M	15	23.3%	6.2
3/3k	3.02M	15	22.8%	6.3
3/4k	3.21M	15	22.7%	6.4
3/5k	3.35M	15	22.6%	6.6
4/2k	2.85M	15	23.2%	6.5
4/3k	3.15M	15	23.0%	6.7
4/4k	3.33M	15	22.6%	6.8
4/5k	3.54M	15	22.5%	7.1
5/2k	3.38M	15	23.2%	7.1
5/3k	3.93M	15	22.7%	8.0
5/4k	4.38M	15	22.5%	8.8
5/5k	4.75M	15	22.1%	9.9
5/3k	346k/816k	15	24.8%	0.99
5/4k	426k/816k	15	24.6%	1.04
5/5k	499k/816k	15	24.3%	1.08

Tab. 3 Unfactored full network sizes, best LM weights, error rates and real-time factors for models with context order K and S states on CSJ test set 1. Size of networks in last three rows is listed separately for $\tilde{H}C \circ L$ and G because they are used here in online-composition mode.

5. Summary

We showed how the direct construction of the transducer mapping sequences of observation state IDs to sequences of phones from the phonetic decision trees of a tied-state acoustic model can lead to a simpler and faster compilation procedure for the final transducers needed for decoding in speech recognition. The proposed algorithm makes it easy to use high-order full cross-word context-dependent models in a one-pass time-synchronous search with little overhead compared to the regularly used triphones, which used to be cumbersome or even impossible for high-order context-dependent models because of memory constraints when the standard method was used.

In our case, the new algorithm resulted not only in a significant simplification of the compilation algorithm, but also in much smaller and therefore more efficient final networks. We showed recognition results for a large Japanese spontaneous speech database (CSJ corpus), which can be decoded in real-time in less then 125 MB total memory using full cross-word quinphones at about 9% search error in a one-pass search.

6. Acknowledgments

We would like to thank the MIT Spoken Language Systems Group for providing some of the basic WFST tools.

文　献

[1] A. Ljolje, D. Hindle, M.D. Riley, and R.W. Sproat. The AT & T LVCSR-2000 system. In *Proceedings of the 2000 Speech Transcription Workshop*, Maryland, 2000.

[2] T. Hain, P.C. Woodland, G. Evermann, and Povey D. New features in the CU-HTK system for the transcription of conversational telephone speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, Salt Lake City, Utah, 2001.

[3] M. Mohri, F. Pereira, and M. Riley. Weighted finite state transducers in speech recognition. In *Proceedings of the Automatic Speech Recognition Workshop*, pages 97–106, Paris, France, 2000.

[4] M. Mohri, M. Riley, D. Hindle, A. Ljolje, and F. Pereira. Full expansion of context-dependent networks in large vocabulary speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume II, pages 665–668, Seattle, 1998.

[5] M. Riley, F. Pereira, and M. Mohri. Transducer composition for context-dependent network expansion. In *Proceedings of the European Conference on Speech Communication and Technology*, volume III, pages 1427–1430, Rhodos, Greece, 1997.

[6] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui. Benchmark test for speech recognition using the corpus of spontaneous japanese. In *Proceedings of the Spontaneous Speech Processing & Recognition Workshop*, pages 135–138, Tokyo, Japan, 2003.

[7] J. J. Odell. *The Use of Context in Large Vocabulary Speech Recognition*. PhD thesis, Cambridge University, Cambridge, England, 1995.

[8] T. Hori, C. Hori, and Y. Minami. Fast on-the-fly composition for weighted finite-state transducers in 1.8 million-word vocabulary continuous speech recognition. In *Proceedings of the International Conference on Spoken Language Processing*, page to appear, Jeju, Korea, 2004.