4th International Conference on Spoken
Language Processing (ICSLP 96)
Philadelphia, PA, USA
October 3-6, 1996

LR-PARSER-DRIVEN VITERBI SEARCH WITH HYPOTHESES MERGING MECHANISM USING CONTEXT-DEPENDENT PHONE MODELS

Tomokazu Yamada, Shigeki Sagayama

NTT Human Interface Laboratories
1-2356, Take, Yokosuka-shi, Kanagawa, 238-03 Japan
e-mail: tomokazu@nttspch.hil.ntt.jp, saga@nttspch.hil.ntt.jp

ABSTRACT

This paper describes a Viterbi search algorithm for continuous speech recognition using context-dependent phone models under the constraint defined by a context-free grammar (CFG). It is based on a frame synchronous LR parser which dynamically generates a finite state network (FSN) from the CFG with an efficient path merging mechanism. Full context-dependency (intra- and inter-word context) is taken into account in the likelihood calculation process.

This paper first describes the algorithm and the processing mechanism, then compares the experimental results of our algorithm and the conventional tree-based HMM-LR speech recognition algorithm which uses HMMs and an LR parser in phone-synchronous processing. The experiments show that our algorithm runs faster than the conventional HMM-LR algorithm with an equivalent recognition accuracy.

1. INTRODUCTION

The combination of a Finite-State Network (FSN) and one-pass search algorithm [4] is suitable for real-time processing of speech recognition because of its frame synchronous processing and ease of implementation. On the other hand, use of Context-Free Grammar (CFG) is advantageous in representing more generalized language constraints. Generalized LR parser [7] is one of most efficient parsers and is often used in natural language processing as well as in continuous speech recognition. It was combined with HMM phone models [1] first in the level-building style and it provided one of the most accurate speech recognition methods producing N-best candidates under a constraint defined by a CFG. This method, however, tends to be inefficient in cases of a large CFG as numerous similar hypotheses may be generated in the tree search process. To cope with this problem, FSN-based one-pass network search with merged hypotheses seems effective. Since, apparently, it may not be feasible to compile an FSN beforehand for the given CFG because of the infinite expansion of recursive grammatical rules, the FSN must be only partially expanded and incrementally and dynamically generated in the search process. From this motivation, several methods integrating FSN-based one-pass search and LR parser have already been proposed incorporating context-dependent phone models and temporary wild-card models [2], context-independent phone models [3], or context-dependent phone models and two types of hypotheses (grid hypotheses and grammartical hypotheses) [5].

This paper proposes an algorithm that is also based on FSN-based one-pass search algorithm and has three major advantages: use of context-free grammar (CFG) for language constraints; dynamic FSN generation; and full context-dependency (intra- and inter-word context) for acoustical models. For dynamic FSN generation taking into consideration full context-dependency and path merging based on LR parsing, we propose a new method called **Delayed Arc Evaluation** which easily locates an appropriate context-dependent phone model, the label of which is given to an FSN arc. The context-dependent phone HMM used here is an HMnet (hidden Markov network) [6] that represents a large variety of allophonic variations in a compact network structure of hidden Markov states.

2. SEARCH ALGORITHM

2.1. Path Merging Using LR Parser

By using a simple Japanese phrase grammar (Figure 1) and a partial hypotheses derived from the grammar (Figure 2), we describe how hypotheses can be merged. This grammar is compiled to an LR table beforehand. The hypothesis generation starts from one hypothesis whose stack has only a state of '0' and then extends the hypotheses corresponding to the possible LR actions described in the table. The stack content of each hypothesis changes by pushing a state to the stack or popping states off the stack according to the action. The numbers shown in Figure 2 represent the stack contents of hypotheses (state numbers). The path is divided into three paths because there are three words for town in this grammar but the paths reach the same node whose stack content is [0, 1, 3] through the reduce action using production rules (2), (3), (4). Therefore, these three paths can be merged at this node.

(0)	**S**	→	- **phrase** -
(1)	**phrase**	→	**town particle**
(2)	**town**	→	k o o f u
(3)	**town**	→	k o o b e
(4)	**town**	→	k o g a n e i
(5)	**particle**	→	k a r a

Figure 1: Japanese phrase grammar

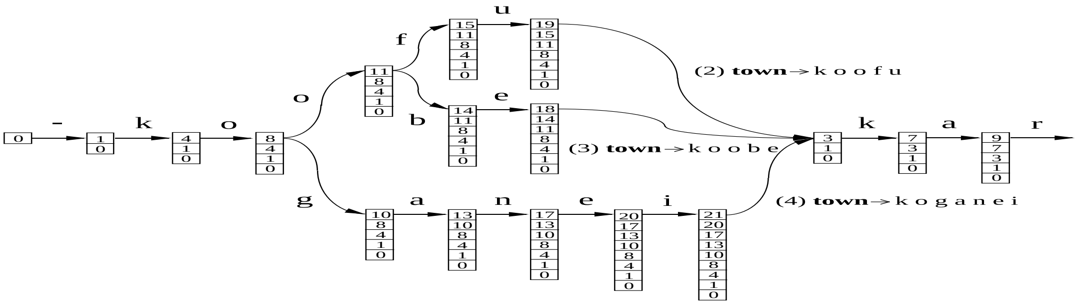

Figure 2: Example of path merging based on LR parsing

2.2. Dynamic FSN Generation Based on Delayed Arc Evaluation

As mentioned above, we can find the merging points based on LR parsing. It is, however, not a simple problem particularly when the phone models are context-dependent and full phonetic context dependency (i.e., both intra-word and cross-word context dependency) is taken into account. The advantage of our algorithm is that it efficiently performs frame synchronous processing with full phonetic context dependency and that it attains a high hypotheses merging efficiency.

In Figure 1, terminal symbols are context-independent phones. If these terminals were context-dependenet phones, the grammar could handle context-dependent phone models. However, in such a case, three other non-terminal symbols would be used respectively instead of the non-terminal symbols **town** and **particle**. Therefore, the merging point such as in Figure 2 would not be obtained (Three paths would exist in parallel for a certain time). If separate non-terminal symbols were not used, it would be necessary to use a device such as context-independent phones for the beginnings and endings of the words.

We decided to use context-independent phones for the CFG terminal symbols and to set the context-dependent phones dynamically every time the network is expanded. This is described more concretely in Figure 3. Here triphones are used as context-dependent models. Basically, FSN nodes are created so that they correspond to the stacks shown in Figure 2. These nodes are also used as the hypotheses for LR parsing. First, for an active node (hereinafter called the source node), a possible LR action is taken and a new stack content is obtained. If there already exists a node whose stack content is the same as the one described above, a new node is not generated but rather a new path is generated (if there is no such a path). In this case, the path is merged with other paths at the node. Otherwise, a new node is generated and the path between the source and the new node is generated. Paths are shown in Figure 3 as dotted lines. There is a context-independent phone corresponding to the path generated by the shift action. (Paths generated by the reduce action has no phones and a null-transition arc is generated between the nodes.) Next, the context-dependent phone model labeled as an input arc of the new node is determined by using a context-independent phone (the right hand context) and both center and left phones of the triphone for each input arc of the source node. Finally, the new arcs labeled with triphones are generated (solid lines in Figure 3.) If the input arc is a null-transition arc, the source node of the null-transition arc is checked recursively and the new arcs are set directly between the nodes.

As the evaluation of a predicted phone as a center phone is delayed until the next step, we call this method "Delayed Arc Evaluation (DAE)".

Merging nodes such as **M** in Figure 3 are generated using this method. They should not be merged necessarily. This problem can be avoided by giving the restriction in the calculation or creating another node between the related input and output arcs. It is also possible to leave this just as it is and to use it as an approximation calculation.

The dynamic FSN generation algorithm based on DAE is summarized as follows:

A	Set of active nodes
ReduceNode(r, n)	New node generated by the reduce action for node n with rule r
ShiftNode(p, s, n)	New node generated by shift action for node n with phone p and push s
ExistNode(n)	If there already exists a node whose stack contents are the same as that of n, return the node
SourceNode(a)	Returns the source node of arc a
ExistConnection(n, m, p)	TRUE if there already exists a connection from node n to m with terminal symbol p (otherwise FALSE)
MakeLRConnection(n, m, p)	Create a connection from node n to m with terminal symbol p
Model(a)	Returns the context-dependent model of the arc a
ExistArc(n, m, q)	TRUE if there already exists an arc from node n to m with context-dependent model q (otherwise FALSE)

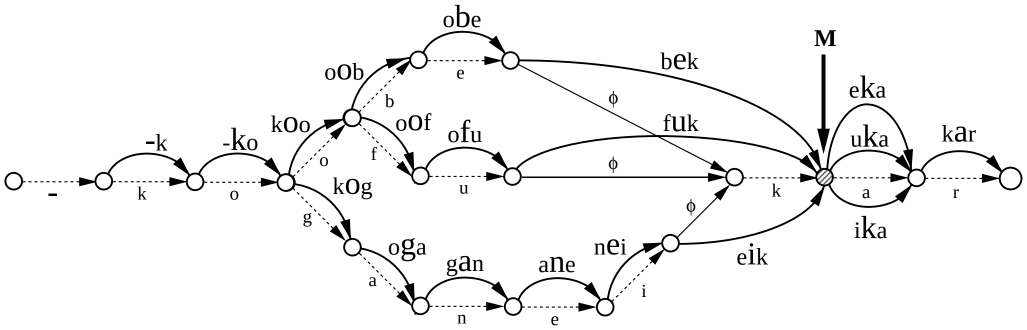

Figure 3: Context-dependent phone-labeled arcs of the FSN

Dynamic FSN Generation Based on DAE

```
$N \leftarrow A$, $S \leftarrow \phi$, $R \leftarrow \phi$
$A \leftarrow \phi$
while $N \neq \phi$ do
    for each node $n \in N$ do
        for each possible LR action $x$ for $n$ do
            if $x$ is reduce action then
                $R \leftarrow R \cup < r, n >$
                where $r$ is applied rule number
            else if $x$ is shift action then
                $S \leftarrow S \cup < p, s, n >$
                where $p$ is the shifted CI-phone,
                $s$ is the LR state number
            end
        end
        remove $n$ from $N$
        $A \leftarrow A \cup \{n\}$
    end
    if $R \neq \phi$ then
        call MakeReduceNodes
    end
end
call MakeShiftNodes
```

MakeReduceNodes

```
$N \leftarrow \phi$
for each item $< r, n > \in R$ do
    $n' \leftarrow$ ReduceNode($r$, $n$)
    $m \leftarrow$ ExistNode($n'$)
    if $m \neq \phi$ then
        $N \leftarrow N \cup \{m\}$
        if not ExistConnection($n$, $m$, $\phi$) then
            MakeLRConnection($n$, $m$, $\phi$)
        end
    else
        MakeLRConnection($n$, $n'$, $\phi$)
        $N \leftarrow N \cup \{n'\}$
    end
end
```

MakeShiftNodes

```
$A \leftarrow \phi$
for each item $< p, s, n > \in S$ do
    $n' \leftarrow$ ShiftNode($p$, $s$, $n$)
    $m \leftarrow$ ExistNode($n'$)
    if $m \neq \phi$ then
        $n' \leftarrow m$
    end
    call SetArcs with item $< n, n', p >$
end
```

SetArcs(n, n', p)

```
for each input-arc $a$ of $n$ do
    if $a \neq$ null-transition then
        $q \leftarrow$ Model($a$)
        $q' \leftarrow$ GetModel($q$, $p$)
        if not ExistArc($n$, $n'$, $q'$) then
            set arc from node $n$ to $n'$
            with context-dependent model $q'$
        end
    else
        $n'' \leftarrow$ SourceNode($a$)
        call SetArcs with item $< n, n'', p >$
    end
end
$A \leftarrow A \cup n'$
```

2.3. Recognition Algorithm

The recognition algorithm based on a one-pass Viterbi search using dynamic FSN generation is specified as follows:

(1) FSN pre-generation and buffer initialization
(2) for each frame, do steps (3) through (5)
(3) Dynamic FSN generation based on DAE
(4) FSN based one-pass Viterbi search
(5) Set active nodes
(6) Backtrace

FSN pre-generation is necessary because arcs labeled with context-dependent phones are not generated during the first few steps.

3. EXPERIMENTS

This section shows the results of the experimental comparison between the proposed algorithm and the conventional HMM-LR algorithm. The experiments were performed using utterances consisting of city names with unnecessary filler uttererances included. The basic form of the utterances is:

sentence → pre_garbage cityname post_garbage.

No acoustic garbage models were used and the garbage is represented as phone sequences in the grammar. There are 88 city names, 83 pre-garbage and 28 post-garbage sequences. The total number of test-set utterances is 352 (88 for 4 speakers).

The speaker-independent context-dependent phone HMM used here is an HMnet [6] and was trained using the ATR word database and the ASJ sentence database. Speech data was sampled at 12 kHz and analyzed using a 32-ms frame length and 8-ms frame shift. The feature vector had 16 cepstrum coeffients, 16 delta coefficients and delta power.

The results are shown in Figure 4 where FSN-LR denotes the proposed algorithm. Here the HMM-LR algorithm may also be extended to generate the hypotheses using a technique which is similar to DAE taking into consideration full context-dependency. Both algorithms have beam search pruning functions, however the method was different, so they are compared based on the processing time. The performance of the HMM-LR algorithm was almost saturated at about an 85% word accuracy. The processing time necessary to obtain the same word accuracy with the proposed algorithm was about 30% of the HMM-LR algorithm required. Furthermore, the proposed algorithm attained higher word accuracy for the top score.

4. CONCLUSION

This paper has proposed an LR-parser-based novel Viterbi search algorithm for continuous speech recognition using context-dependent phone models under the constraint defined by CFG. The Delayed Arc Evaluation method has also been proposed for dynamically generated FSN which takes into consideration full context-dependency and effective path merging based on LR parsing. The experimental results show that our algorithm requires less computation than the normal HMM-LR algorithm does.

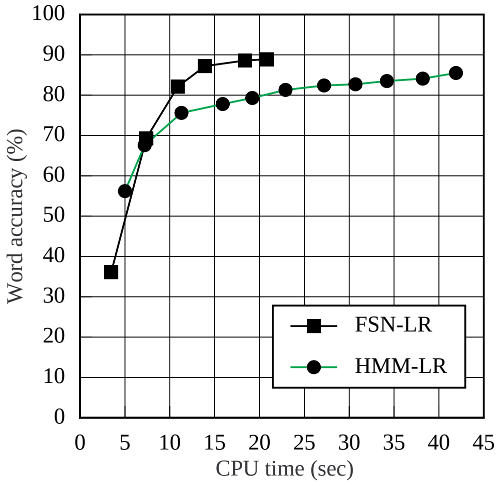

Figure 4: Word accuracy as a function of the average CPU time

ACKNOWLEDGEMENTS

The authors would like to thank Mr. Satoshi Takahashi who provided the acoustic models.

REFERENCES

[1] T. Hanazawa, K. Kita, S. Nakamura, T. Kawabata, and K. Shikano, "ATR HMM-LR continuous speech recognition system," In *ICASSP*, 2.4, 1990.

[2] K. Itou, S. Hayamizu, and H. Tanaka, "Continuous speech recognition by context dependent phonetic HMM and an efficient algorithm for finding n-best sentence hypotheses," In *ICASSP92*, 10.6, 21–24, 1992.

[3] K. Kita, Y. Yano, and T. Morimoto, "One-pass continuous speech recognition directed by generalized LR parsing," In *ICSLP*, 1.4, 13–16, 1994.

[4] H. Ney, "The use of a one-stage dynamic programming algorithm for connected word recognition," *IEEE Trans. Acoustics, Speech and Signal Processing*, ASSP-32(2): 263–271, 1984.

[5] T. Shimizu, S. Monzen, H. Singer, and S. Matsunaga, "Time-synchronous continuous speech recognizer driven by a context-free grammar," In *ICASSP95*, RP02.07, 584–587, 1995.

[6] J. Takami and S. Sagayama, "A successive state splitting algorithm for efficient allophone modeling," In *ICASSP*, 66.6, 573–576, 1992.

[7] M. Tomita, *Efficient Parsing for Natural Language*, Kluwer Academic Publishers, 1986.