Parsing Continuous Speech by HMM-LR Method

Kenji KITA, Takeshi KAWABATA, Hiroaki SAITO

ATR Interpreting Telephony Research Laboratories
Seika-chou, Souraku-gun, Kyoto 619-02, JAPAN

Abstract

This paper describes a speech parsing method called HMM-LR. In HMM-LR, an LR parsing table is used to predict phones in speech input, and the system drives an HMM-based speech recognizer directly without any intervening structures such as a phone lattice. Very accurate, efficient speech parsing is achieved through the integrated processes of speech recognition and language analysis. The HMM-LR method is applied to large-vocabulary speaker-dependent Japanese phrase recognition. The recognition rate is 87.1% for the top candidates and 97.7% for the five best candidates.

1 Introduction

This paper describes a speech parsing method called HMM-LR. This method uses an efficient parsing mechanism, a generalized LR parser, driving an HMM-based speech recognizer directly without any intervening structures such as a phone lattice.

Generalized LR parsing [1] is a kind of LR parsing [2], originally developed for programming languages and has been extended to handle arbitrary context-free grammars. An LR parser is guided by an LR parsing table automatically created from context-free grammar rules, and proceeds left-to-right without backtracking. Compared with other parsing algorithms such as the CYK (Cocke-Younger-Kasami) algorithm [3] or Earley's algorithm [4], a generalized LR parsing algorithm is the most efficient algorithm for natural language grammars.

There have been some applications of generalized LR parsing to speech recognition. Tomita [5] proposes an efficient word lattice parsing algorithm. Saito [6] proposes a method of parsing phoneme sequences that include altered, missing and/or extra phonemes. However, these methods are inadequate because of the information loss due to signal-symbol conversion. The HMM-LR method does not use any intervening structures. The system drives an HMM-based speech recognizer directly for detecting/verifying phones predicted using an LR parsing table.

HMM (Hidden Markov Models) [7] has the ability to cope with the acoustical variation of speech by means of stochastic modeling, and it has been used widely for speech recognition. In HMM, any word models can be composed of phone models. Thus, it is easy to construct a large vocabulary speech recognition system.

This paper is organized as follows. Section 2 describes the LR parsing mechanism. Section 3 describes HMM. Section 4 describes the HMM-LR method. Section 5 describes recognition experiments using HMM-LR. Finally, section 6 presents our conclusions.

2 LR Parsing

2.1 LR Parsing

LR parsing was originally developed for programming languages. It is applicable to a large class of context-free grammars.

The LR parser is deterministically guided by an LR parsing table with two subtables (*action table* and *goto table*). The action table determines the next parser action $ACTION[s,a]$ from the state s currently on top of the stack and the current input symbol a. There are four kinds of actions, *shift*, *reduce*, *accept* and *error*. *Shift* means shift one word from input buffer onto the stack, *reduce* means reduce constituents on the stack using the grammar rule, *accept* means input is accepted by the grammar, and *error* means input is not accepted by the grammar. The goto table determines the next parser state $GOTO[s,A]$ from the state s and the grammar symbol A.

The LR parsing algorithm is summarized below.

1. *Initialization*. Set p to point to the first symbol of the input. Push the initial state 0 on top of the stack.
2. Consult $ACTION[s,a]$ where s is the state on top of the stack and a is the symbol pointed to by p.
3. If $ACTION[s,a]$ = "*shift* s'", push s' on top of the stack and advance p to the next input symbol.
4. If $ACTION[s,a]$ = "*reduce* $A \rightarrow \beta$", pop $|\beta|$ symbols off the stack and push $GOTO[s',A]$ where s' is the state now on top of the stack.
5. If $ACTION[s,a]$ = "*accept*", parsing is completed.
6. If $ACTION[s,a]$ = "*error*", parsing fails.
7. Return to 2.

2.2 Generalized LR Parsing

Standard LR parsing cannot handle ambiguous grammars. For an ambiguous grammar, the LR parsing table will have *multiple entries* (*conflicts*). As a general method, a stack-splitting mechanism can be used to cope with multiple entries. Whenever a multiple entry is encountered, the stack is divided into two stacks, and each stack is processed in parallel. Thus, it is possible to use LR parsing to handle an ambiguous grammar which describes natural language.

(1) S → NP V
(2) S → V
(3) NP → N
(4) NP → N P
(5) N → k o r e
(6) P → o
(7) V → k u r e
(8) V → o k u r e

Fig.1 Example grammar

	e	o	u	k	r	$	S	N	V	P	NP
0		s5		s2			6	1	3		4
1		s7,r3		r3						8	
2		s9	s10								
3						r2					
4		s5		s11					12		
5				s13							
6						acc					
7				r6							
8				r4							
9					s14						
10					s15						
11			s10								
12						r1					
13			s16								
14	s17										
15	s18										
16					s19						
17		r5		r5							
18						r7					
19	s20										
20						r8					

Fig.2 LR parsing table

A simple example grammar is shown in Fig.1, and the LR parsing table, compiled from the grammar automatically, is shown in Fig.2. The left part is the action table and the right part is the goto table. The entry "acc" stands for the action "*accept*", and blank spaces represent "*error*". The terminal symbol "$" represents the end of the input.

3. HMM (Hidden Markov Models)

HMM is effective in expressing speech statistically, so it has been used widely for speech recognition.

Fig.3 shows an example of a phone model. A model has a collection of *states* connected by *transitions*. Two sets of probabilities are attached to each transition. One is a *transition probability* a_{ij}, which provides the probability for taking transition from state i to state j. The other is an *output probability* b_{ijk}, which provides the probability of emitting code k when taking a transition from state i to state j.

The *forward-backward algorithm* [7] can be used to estimate the model's parameters given a collection of training data. After estimating the model's parameters, the *forward algorithm* (*trellis algorithm*) can be used to verify phones as follows.

$$\alpha_i(t) = \begin{cases} 1 & (t = 0 \ \& \ i = 0) \\ 0 & ((t = 0 \ \& \ i \neq 0) \ or \ (t \neq 0 \ \& \ i = 0)) \\ \sum_j (\alpha_j(t\text{-}1) a_{ji} b_{ji}(y_t)) \end{cases}$$

$\alpha_i(t)$ is the probability that the Markov process is in state i having generated code sequence $y_1, y_2, \ldots, y_t$. The final probability for the phone is given by $\alpha_F(T)$ where F is a final state of the phone model and T is a length of input code sequence.

4. HMM-LR Method

4.1 Basic Mechanism

In standard LR parsing, the next parser action (shift, reduce, accept or error) is determined using the current parser state and next input symbol to check the LR parsing table. This parsing mechanism is valid only for symbolic data and cannot be applied simply to continuous data such as speech.

In HMM-LR, the LR parsing table is used to predict the next phone in the speech. For the phone prediction, the grammar terminal symbols are phones instead of the grammatical category names generally used in natural language processing. Consequently, a lexicon for the specified task is embedded in the grammar.

The following describes the basic mechanism of HMM-LR (see Fig.4). First, the parser picks up all phones which the initial state of the LR parsing table predicts, and invokes the HMM to verify the existence of these predicted phones. During this process, all possible parsing trees are constructed in

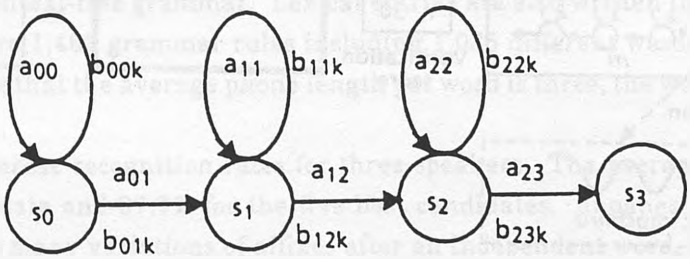

Fig. 3 HMM phone model

parallel. The HMM phone verifier receives a probability array which includes end point candidates and their probabilities, and updates it using an HMM probability calculation process (the forward algorithm). This probability array is attached to each partial parsing tree. When the highest probability in the array is lower than a threshold level, the partial parsing tree is pruned by threshold level, and also by beam-search technique. The parsing process proceeds in this way, and stops if the parser detects an accept action in the LR parsing table. In this case, if the best probability point reaches the end of speech data, parsing ends successfully. A very accurate, efficient parsing method is achieved through the integrated process of speech recognition and language analysis. Moreover, HMM units are phones, and any word models can be composed of phone models, so it is easy to construct a large vocabulary speech recognition system.

4.2 Algorithm

To describe an algorithm for the HMM-LR method, we first introduce a data structure named *cell*. A cell is a structure with information about one possible parsing. The following are kept in the cell:

- LR stack, with information for parsing control.
- Probability array, which includes end point candidates and their probabilities.

The algorithm is summarized below.

1. *Initialization.* Create a new cell C. Push the LR initial state 0 on top of the LR stack of C. Initialize the probability array Q of C;

$$Q(t) = \begin{cases} 1 & t = 0 \\ 0 & 1 \leqq t \leqq T \end{cases}$$

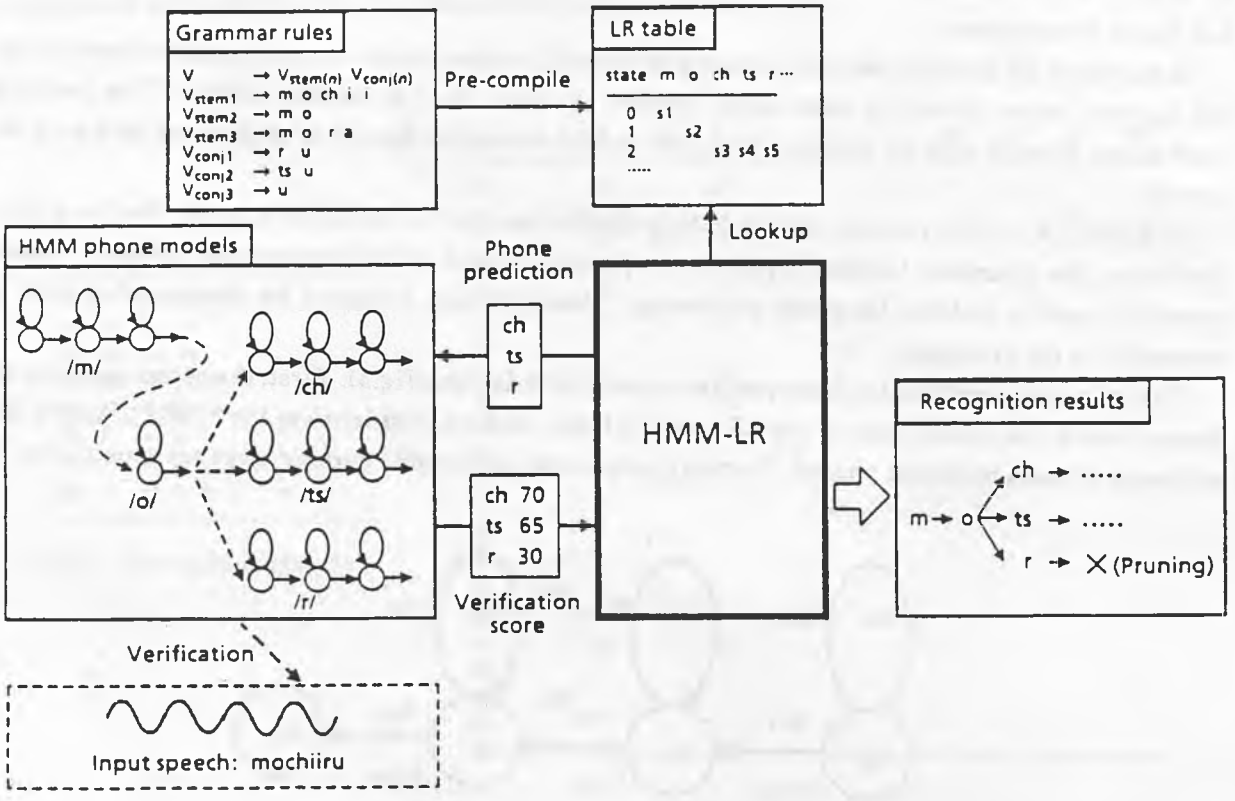

Fig. 4 Basic mechanism of HMM-LR

2. *Ramification of cells.* Construct a set

$$S = \{ (C, s, a, x) \mid \exists C, a, x \,(\, C \text{ is a cell } \&\ C \text{ is not accepted} \\ \&\ s \text{ is a state of } C\ \&\ ACTION[s,a] = x\ \&\ x \neq \text{"error"} \}.$$

For each element $(C, s, a, x) \in S$, do operations below. If a set S is empty, parsing is completed.

3. If $x =$ "*shift s'*", verify the existence of phone a. In this case, update the probability array Q of the cell C by the following computation.

$$a_i(t) = \begin{cases} Q(t) & (i = 0) \\ 0 & (t = 0\ \&\ i \neq 0) \\ \Sigma_j (a_j(t\text{-}1) a_{ji} b_{ji}(y_t)) \end{cases}$$

$$Q(t) = \begin{cases} 0 & (t = 0) \\ a_F(t) \end{cases}$$

If $max\ Q(i)$ $(i = 1...T)$ is below a threshold level set in advance, the cell C is abandoned. Else push s' on top of the LR stack of the cell C.

4. If $x =$ "*reduce* $A \rightarrow \beta$", same as standard LR parsing.
5. If $x =$ "*accept*" and $Q(T)$ is larger than a threshold level, the cell C is accepted. If not, cell C is abandoned.
6. Return to 2.

Recognition results are kept in cells. Generally, many recognition candidates exist, and it is possible to rank these candidates using a value $Q(T)$.

The set S constructed in step 2 above is quite large. It is possible to set an upper limit on the number of elements in S by beam-search technique. It is also possible to use *local ambiguity packing* [1] to represent cells efficiently.

5. Experiments

The HMM-LR method is applied to speaker-dependent Japanese phrase recognition. Duration control techniques and separate vector quantization are used to achieve accurate phone recognition. Two duration control techniques are used, one is phone duration control for each HMM phone model and the other is state duration control for each HMM state [8]. Phone duration control is carried out by weighting HMM output probabilities with phone duration histograms obtained from training sample statistics. State duration control is realized by state duration penalties calculated by modified forward-backward probabilities of training samples. In separate vector quantization, spectral features, spectral dynamic features and energy are quantized separately. In the training stage, the output vector probabilities of these three codebooks are estimated simultaneously and independently, and in the recognition stage all the output probabilities are calculated as a product of the output vector probabilities in these codebooks.

The grammar used in the experiments describes a general Japanese syntax of phrases and is written in the form of context-free grammar. Lexical entries are also written in the form of context-free grammar. There are 1,461 grammar rules including 1,035 different words, and perplexity per phone is 5.87. Assuming that the average phone length per word is three, the word perplexity is more than 100.

Table 1 shows the phrase recognition rates for three speakers. The average recognition rate is 87.1% for the top candidate and 97.7% for the five best candidates. Japanese is an agglutinative language, and there are many variations of affixes after an independent word. The problem here is that recognition errors are often mistakes caused by these affixes.

Table 1 Phrase recognition rates

Order \ Speaker	MAU	MHT	MNM	Average
1	87.8	85.6	87.8	87.1
2	98.6	93.5	92.8	95.0
3	99.3	96.8	95.0	97.0
4	99.3	97.5	95.7	97.5
5	99.6	97.8	95.7	97.7

6. Conclusion

In this paper, we described a speech parsing method called HMM-LR, which uses a generalized LR parsing mechanism and an HMM-based speech recognizer. The experiment results show that an HMM-LR method is very effective in continuous speech recognition.

An HMM-LR continuous speech recognition system is used as part of the SL-TRANS (Spoken Language TRANSlation) system developed at ATR Interpreting Telephony Research Laboratories.

Acknowledgements

The authors would like to express their gratitude to Dr.Akira Kurematsu, president of ATR Interpreting Telephony Research Laboratories, for his encouragement and support, which made this research possible, and to Mr.Toshiyuki Hanazawa for the HMM program.

References

[1] Tomita, M.: *Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems*, Kluwer Academic Publishers (1986).

[2] Aho, A.V., Sethi, R. and Ullman, J.D.: *Compilers, Principles, Techniques, and Tools*, Addison-Wesley (1986).

[3] Aho, A.V. and Ullman, J.D.: *The Theory of Parsing, Tranlation, and Compiling*, Prentice-Hall, Englewood Cliffs (1972).

[4] Earley, J.: *An Efficient Context-Free Parsing Algorithm*, Comm. ACM, Vol.13, No.2, pp.94-102 (1970).

[5] Tomita, M.: *An Efficient Word Lattice Parsing Algorithm for Continuous Speech Recognition*, Proc. IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP-86, pp.1569-1572 (1986).

[6] Saito, H. and Tomita, M.: *Parsing Noisy Sentences*, Proc. 12th Int. Conf. Comput. Linguist. COLING-88, pp.561-566 (1988)

[7] Levinson, S.E., Rabiner, L.R. and Sondhi, M.M.: *An Introduction to the Application of the Theory of Probabilistic Functions of a Markov Process to Automatic Speech Recognition*, Bell Syst. Tech. J., Vol.62, No.4, pp.1035-1074 (1983).

[8] Hanazawa, T., Kawabata, T. and Shikano, K.: *Duration Control Methods for HMM Phoneme Recognition*, The Second Joint Meeting of ASA and ASJ (1988).