Integration of Phonological and Linguistic Information for Speech Recognition

TANAKA Hozumi, TOKUNAGA Takenobu and LI Hui
Department of Computer Science
Tokyo Institute of Technology
2-12-1 Ookayama Meguro Tokyo 152 Japan
e-mail: tanaka@cs.titech.ac.jp

1. Introduction

It is obvious that successful speech recognition requires the use of linguistic information. To this end, a generalized LR (GLR) parser provides an exceptionally good tool, as it combines in a very flexible way linguistic and phonological information.

One problem with continuous speech recognition of real world texts containing large vocabulary is the reduction of the search space. GLR parsers can meet this requirement[kita:89:a] by using linguistic constraints. In the phone-based speech recognition system, the GLR parser has been used as a phoneme predictor. The GLR parser[tomita:86:a] is guided by an LR table, automatically generated from context-free grammar (CFG) rules. It proceeds left-to-right without backtracking. In order to make phone predictions, the lookahead symbols in LR table are phones rather than of the usual grammatical categories. Lexical rules are thus expressed as follows:

<grammatical category> -> <sequence of phones>.

Several experiments[itou:92:a,lee:89:a,nagai:93:a,schwartz:85:a] have shown that the performance of speech recognition systems could be improved by using allophones rather than phones as recognition units. Allophone models (such as triphone models) are context-dependent phone models that take into consideration both the left and right neighboring phones to model the major coarticulatory effects in continuous speech.

The combination of allophone models and a GLR parser[itou:92:a,nagai:93:a] is desirable in order to achieve better performance in continuous speech recognition. The main problem of integrating GLR parsing into an allophone-based recognition system is how to solve the word juncture problem, that is, how to express the phones at a word boundary with allophone models.

In this paper, we propose a new method to generate an allophone-based LR table that can solve the word juncture problem., By combining allophone rules with syntactic and lexical rules, this method generates an LR table, which it modifies then on the basis of an allophone connection matrix by using constraint propagation methods(CPM).

The organization of this paper is as follows. Section 2 provides an overview of the past allophone-based GLR parsing methods and points out problems in those methods. After discussing the advantages of using the canonical LR table for speech recognition, section 3 describes our method to generate an allophone-based LR table by applying CPM. Section 4 compares LR tables before and after having applied CPMs; and Section 5 points towards future developments.

In the following sections, we will use several examples from Thai . It should be noted, however, that the method proposed is not language specific, - in principal it can be applied to any spoken language.

2. Overview of Allophone-Based GLR Parsing Method

The main problem of integrating GLR parsing into an allophone-based recognition system is solving the word juncture problem (segmentation). Consider a Japanese word "a k i(autumn)", which is a sequence of three phones in the lexical rule that follows:

adj -> a k i

The allophone corresponding to the phone "k", can be determined by the left and right context which are known in advance, namely "a" and "i" respectively. Let this phone "k" be an allophone "k1." The phones "a" and "i" are located at the word boundary. For the beginning phone "a" there is no left context, and for the final phone "i", there is no right context. Hence for the phones located at the word boundaries, it is difficult to anticipate their respective left or right contexts. We call this the word juncture problem. Consequently we get

adj -> a k1 i.

3. Algorithm for Generating Allophone-based LR Table through Constraint Propagation

In this section we propose a new method called CPM (Constraint Propagation Method) in order to generate a phoneme-context-dependent(allophone-based) LR table. This method allows to solve the word juncture problem and to predict allophones.

The outline of our method is shown in Fig. 1.

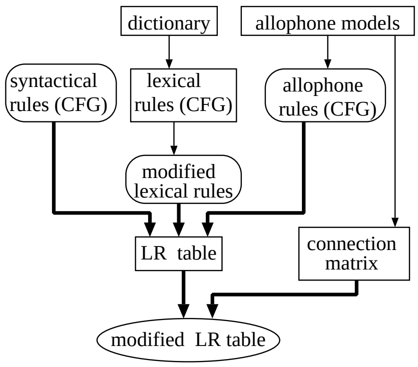

Fig.1: Outline of CPM

In this method, , an allophone connection matrix and a set of allophone rules are constructed on the basis of an allophone model. Lexical rules determine how word intrinsic phones are changed into allophones. From a set of syntactic, lexical and allophone rules(CFG), an initial allophone-based LR table is generated, which is then modified by using the allophone connection matrix and constraint propagation methods.

Before explaining the details of each step, we would like to discuss the types of LR tables. All methods mentioned in Section 2 have used the SLR or LALR table. Compared with the canonical LR table, the SLR and LALR table can not provide the precise phoneme predictions because the SLR and LALR table have fewer states due to merging of several states in an LR table[aho:86:a]: yet merging several states transforms many actions into a state in which they produce many predictions. This is why our method uses the canonical

LR table. Generally, the canonical LR table has more states than the SLR or LALR table, but by applying CPM, we can achieve substantial reduction of the table's size.

In this section, we will illustrate our method by using an example of simple syntactic and lexical rules(CFG) of Thai. See - Fig. 2.

(1)	S $\rightarrow$ N BE	(3)	N $\rightarrow$ ch i ch i (father)
(2)	N $\rightarrow$ h a h a (mother)	(4)	BE $\rightarrow$ d a (be)

Fig.2: An example of the CFG rules and lexical rules

3.1 Allophone connection matrix

The allophone context of an allophone "x" is defined as:

<left context> x <right context>

where <left context> and <right context> are a set of phones. We assume that there is only one allophone context per allophone. An allophone connection matrix is created from a set of allophone contexts.

Fig. 3 is an example of the allophone connection matrix partially filled. We will use this connection matrix to incorporate allophone connection constraints into the LR table at Section 1 by applying CPM.

LEFT \ RIGHT	h1	h2	a1	a2	ch1	ch2	i1	i2	d1	d2	d3	$
h1			1	0								
h2			0	0								
a1	1	1							1	0	1	0
a2	0	0							0	0	0	1
ch1							0	0				
ch2							0	1				
i1					0	0			0	0	0	
i2					1	1			0	1	1	
d1			0	1								
d2			0	1								
d3			0	0								

Fig.3: An example of the allophone connection matrix

3.2 Modification of the lexical rules

The phones within a word are automatically converted into the allophones. This is done by using allophone contexts. Please note that we can not change the phones at the word boundaries because of the word juncture problem. Therefore, the lexical rules(rules 2 to 4 in the Fig. 2 are changed into those shown in the figure here below:

(2)'	N $\rightarrow$ h a1 h1 a	(4)'	BE $\rightarrow$ d a
(3)'	N $\rightarrow$ ch i2 ch2 i		

Fig.4: Conversion of the lexical rules

3.3 Allophone-based LR table

The allophone rules are derived by associating a phone with the corresponding allophones:

<phone> -> <allophone1> | <allophone2> | ...

Assume the following couples: h1, h2 for "h", a1, a2 for "a", ch1, ch2 for "ch", i1, i2 for "i", d1, d2 for "d", on the basis of these data set of allophone rules (see (5) to (14) in Fig. 5) can be produced.

(5) h → h1	(9) ch → ch1	(13) d → d1
(6) h → h2	(10) ch → ch2	(14) d → d2
(7) a → a1	(11) i → i1	(15) d → d3
(8) a → a2	(12) i → i2	

Fig.5: A set of the allophone rules

Now we have a set of syntactic, lexical and allophonic rules as shown in Fig. 6

(1) S → N BE	(6) h → h2	(11) i → i1
(2)' N → h a1 h1 a	(7) a → a1	(12) i → i2
(3)' N → ch i2 ch2 i	(8) a → a2	(13) d → d1
(4)' BE → d a	(9) ch → ch1	(14) d → d2
(5) h → h1	(10) ch → ch2	(15) d → d3

Fig.6: A set of the CFG, lexical and allophone rules

From these extended CFG rules we can generate a canonical LR table (see Fig. 7). Note that all the lookahead symbols of this table are allophones.

	ACTION											GOTO							
state	h1	h2	a1	a2	ch1	ch2	i1	i2	d1	d2	$	h	a	ch	i	d	N	BE	S
0	sh5	sh6(c)			sh2(c)	sh3						4		1			7		25
1								sh8											
2								re9(a)											
3								re10											
4		(c)	sh9																
5			re5																
6			re6(a)																
7									sh11	sh12(c)						10		13	
8						sh21													
9	sh17																		
10			sh15	sh16(c)									14						
11			re13	re13(a)															
12			re14(a)	re14(a)						(d)									
13											re1								
14						(b)					re4								
15											re7								
16											re8(a)								
17			sh19	sh20(b)									18						
18									re2	re2(d)									
19									re7	re7(d)	(d)								
20									re8(a)	re8(a)									
21							sh23(b)	sh24							22				
22									re3	re3(d)									
23							(f)		re11(f)	re11(a)									
24									re12	re12(a)									
25											acc								

Fig.7: Canonical LR(CLR) table generated from rules in Fig. 6

In the extended CFG rules in Fig. 6, the information about connectability represented in Fig. 3 is not included for the phones at the word boundaries.

3.4 Modifications of LR Table

In order to incorporate allophone connection constraints into the LR table, we combine our new CPM (Constraint Propagation Method) originally developed by Tanaka et al.[tanaka:93:a]. We illustrate the new CPM method below.

Initial constraints are imposed on the LR table by using allophone connection matrix. These constraints propagate throughout the LR table producing a modified allophone-based LR table.

	ACTION						GOTO							
state	h1	a1	ch2	i2	d1	$	h	a	ch	i	d	N	BE	S
0	sh2		sh7				1		5			4		25
1		sh8												
2		re5												
4					sh11						10		9	
5				sh13										
7				re10										
8	sh21													
9						re1								
10		sh19						18						
11		re13												
13			sh14											
14				sh17						15				
15					re3									
17					re12									
18						re4								
19						re7								
21		sh23						22						
22					re2									
23					re7									
25						acc								

Fig.8: The modified canonical LR(MCLR) table after CPM

CPM consists of the following procedures, each of which is executed in this order.

1. Initialization procedure:

   Let Sh be a set of shift actions that are executed immediately before an action A. Delete A with a lookahead symbol w if, for all x with a lookahead symbol v in Sh, connect(v,w)=0.

2. Constraint propagation procedure:

   for each action A in LR table delete A if there is no action executed immediately before or after A.

3. Compaction procedure 1:

   delete row R in the LR table if there is no action in R.

4. Compaction procedure 2:

   delete column C in the LR table if there is no action in C.

Fig. 8 is the modified allophone-based canonical LR(MCLR) table obtained by applying CPM to the table in Fig. 7.

4. The Effects of CPM

CPM enables us to introduce the phoneme-context-dependence into the phones at word boundaries and solve the word juncture problem. Compared with the table of Fig. 7, the lookahead symbols in Fig. 8 include only “a1”, “a2”, “ch2”, “d1”, “d2”, “h1” and “i2”. The

Table 1: Comparison of table size before and after CPM

allophones	table	state	shift	reduce	goto
128	CLR	1722	2260	9254	421
	MCLR	1016	873	1015	421
256	CLR	2959	4760	37831	421
	MCLR	1061	930	1072	421
512	CLR	5627	10211	161127	421
	MCLR	1096	971	1135	421
1024	CLR	11157	21057	701370	421
	MCLR	1139	1003	1212	421

allophones at word boundaries are restricted by CPM through deleting the illegal actions. This very simple example suggests how the word juncture problem can be solved by applying CPM.

For a task with 64 syntactic rules and 120 lexical rules, the canonical LR (CLR) table is compared in terms of the number of states and actions before and after constraint propagation.

The syntactic rules, lexical rules and allophone models we used are the same as in [itou:92:a], which have already been used in a speech recognition system.

Table 1 shows the table's size before and after CPM for 128, 256, 512 and 1024 allophone models. After applying CPM, in the case of 128 allophone models, the number of states, shift actions, and reduce actions decreases to 61.4%, 38.6%, and 11% of the original canonical LR table, and in the case of 1024 allophone models, the number of states, shift actions, and reduce actions decreases to 11%, 4.9%, and 0.2% of the original canonical LR table. The greater the number of allophones are, the greater the benefits in terms the table size shrinking.

Reduction of the reduce and shift actions implies that the allophone predictions in speech recognition become more accurate.

5. Conclusion

We have proposed a new method to generate an allophone-based LR table that solves the word juncture problem. Furthermore it allows us to predict allophones precisely in speech recognition[li:96]. By combining allophone rules with syntactic and lexical rules, an initial LR table is generated. Then, on the basis of an allophone connection matrix the LR table is modified by using CPM.

The characteristics of our approach can be summarized as follows:

(1) Generation of allophone-based LR table is performed at two levels, this enables us to use the existing LR table generation algorithms to generate an initial allophone-based LR table.

(2) By introducing an allophone connection matrix and applying CPM, a large number of illegal states and actions of initial LR table can be deleted, which reduces considerably the tables' size. In addition is solves the word juncture problem. Last but not least, the reduction of actions and states provides us with accurate allophone predictions in speech recognition.

(3) Our method allows to solve the word juncture without any changes in the GLR parsing algorithm: the connection constraints between two adjacent allophones have been incorporated into an LR table.

While our method allows to use existing LR table generation methods in order to get the initial LR table, the number of states and actions of initial LR table often grow exponentially as the number of syntactic and lexical rules or allophone models increases. In order to improve this situation, we can modify the LR table generation algorithm slightly in order to incorporate the allophone connection constraints during the generation process.

Future works will address this problem by incorporating the LR table generated by CPM into an allophone-based continuous speech recognition system.

References

[aho:86:a] A.V. Aho, S. Ravi, and J.D. Ullman. Compilers: Principle, Techniques, and Tools. Addision Wesely, 1986.

[itou:92:a] 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 Proc. ICASSP-92, pages I–21–I–24, 1992.

[kita:89:a] K. Kita, T. Kawabata, and H Saito. HMM continuous speech recognition using predictive LR parsing. In Proc. ICASSP-89, pages 703–706. IEEE, 1989.

[lee:89:a] K. F. Lee. Automatic Speech Recognition : The Development of the SPHINX System. Kluwer Academic Publishers, 1989.

[li:96] Hui Li. Integrating Connection Constraints into a GLR parser and its applications in a continuous speech recognition system. Ph.D. dissertation of Tokyo Institute of Technology, 1996.

[nagai:93:a] A. Nagai, S. Sagayama, K. Kita, and H. Kikuchi. Three different lr parsing algorithms for phoneme-context- dependent HMM based continuous speech recognition. IEICE Trans. Inf. & Syst., E76-D(1):29–37, 1993.

[schwartz:85:a] R. Schwartz, Y. Chow, O. Kimball, S. Roucos, M. Krasner, and J. Makhoul. Context dependent modeling for acoustic phonetic recognition of continuous speech. In Proc. ICASSP-85, pages 1205–1208. IEEE, 1985.

[tanaka:93:a] H. Tanaka, T. Tokunaga, and M. Aizawa. Integration of morphological and syntactic analysis based on LR parsing algorithm. In Proc. International Workshop on Parsing Technologies, pages 101–109, Tilburg, 1993.

[tomita:86:a] M. Tomita. Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publishers, Boston, MA, 1986.