A Structured Statistical Language Model conditioned by Arbitrarily Abstracted Grammatical Categories based on GLR Parsing

Tomoyosi AKIBA and Katunobu ITOU

Information Technology Research Institute
National Institute of Advanced Industrial Science and Technology (former ETL)
t-akiba@aist.go.jp

Abstract

This paper presents a new statistical language model for speech recognition, based on Generalized LR parsing. The proposed model, the Abstracted Probabilistic GLR (APGLR) model, is an extension of the existing structured language model known as the Probabilistic GLR (PGLR) model. It can predict next words from arbitrarily abstracted categories. The APGLR model is also a generalization of the original PGLR model, because PGLR can be considered to be a special case of APGLRs that predict the next words from the least abstracted grammatical categories, namely the terminal symbols. The selection of the abstraction level is arbitrary; we show several strategies to define the level. The experimental results show that the proposed model performs better than the original PGLR model for speech recognition.

1. Introduction

Statistical language models have gained a reputation as providing the overall performance for speech recognition, and so widely used in speech recognition systems today. Although n-gram models have the simplest structure of the various statistical models, they have been widely used in these systems. Several recent studies have tried to incorporate linguistic structures into statistical language models. One of the advantages of such structured models is they can take a long-distance correlation between words into account.

Context Free Grammars (CFGs) have been widely used for modeling natural language and therefore have been commonly adopted as the structure of statistical models. Most methods to introduce statistics into CFGs distribute a probability over production rules. Probabilistic Context Free Grammars (PCFGs) are CFGs having a probability distribution defined over all production rules that share their left-hand side. But since PCFGs have the drawback that they cannot capture context-sensitivity over a rule, there have been various extensions of PCFGs to include context-sensitivity. In NLP, several studies attempted to extend the probabilities to treat the context over a rule[1]. On the other hand, in speech recognition, left-to-right models are preferred because they are compatible with the search procedures of speech recognition. For example, Chelba and Jelinek[2] proposed a language model that computes the probability of the next word based upon the grammatical constituents already processed.

Aside from rule-based models, one of the main alternative statistic models is that based on generalized LR parsing[3]. An LR table is a representation of the CFG-parsing process that can be pre-compiled before parsing, so it can be considered as another representation of CFGs. Briscoe and Carroll [4] proposed the direct distribution of probabilities to each action in an LR table. Inui [5] presented the formalization of probabilistic GLR parsing, which is called the Probabilistic GLR (PGLR) model.

One of the main advantages of a model based on LR parsing is that it can naturally capture context-sensitivity based on the nature of left-to-right parsing, in contrast with rule-based models. It can be used to predict words, given the state of parsing, that correspond to the history of the words. It is also suitable for use with the search procedures of speech recognition. Several speech recognition system based on LR parsing have been developed (for example, [6]), so it should be feasible to introduce PGLR models into these systems. Another significant advantage is that the probabilistic parameters can be easily trained simply by counting the frequency of the application of each action by means of parsing training sentences.

In spite of these advantages of the PGLR model, it has been little discussed relative to the speech recognition. This paper relates the state of the LR table to the history of language models and presents an extension of PGLR models towards language models for speech recognition purpose. Our model is called the Abstracted Probabilistic GLR model (APGLR model), because it can predict words from arbitrarily abstracted grammatical categories. This model can be viewed as a generalization of the PGLR model, since the PGLR model is a special case of the APGLR model that predicts next words from the least abstracted grammatical categories, namely the terminal symbols. The formalization of the APGLR model will be shown in the next section.

2. Formalism

2.1. Probabilistic GLR model

The Probabilistic GLR (PGLR) model is defined as one giving the probability to the actions of the LR table. A GLR parsing process can be seen as the sequences $\sigma_0 l_1 a_1 \sigma_1 \cdots l_n a_n \sigma_n$, in which l_i, a_i, and σ_i are the lookahead symbol (the next terminal symbol), the parser's action, and the parser's state (namely, the parser's stack in case of LR parsing), respectively. Suppose the resulting parse tree T is obtained by the process, then the probability of T is defined as follows.

$$P(T) = P(\sigma_0 l_1 a_1 \sigma_1 l_2 \cdots \sigma_{n-1} l_n a_n \sigma_n) \tag{1}$$

$$\approx P(\sigma_0) \prod_{i=1}^{n} P(l_i a_i \sigma_i | \sigma_{i-1}) \tag{2}$$

The PGLR model[5] approximates the sequence as follows,

$$P(l_i a_i \sigma_i | \sigma_{i-1}) \approx \begin{cases} P(l_i a_i | s_{i-1}) & s_{i-1} \in S_s \\ P(a_i | l_i s_{i-1}) & s_{i-1} \in S_r \end{cases} \tag{3}$$

where s_i is the state of the LR table appeared on the top of the stack σ_i, and S_s and S_r is the set of states immediately after the action 'shift' and 'reduce', respectively.[1]

Since the PGLR model has been studied in NLP, in which the input word sequence can be observed unambiguously, let us reconsider the meaning of the equation (3), in order to use it as a language model for speech recognition. One of the most important objectives of a language model is to predict the next words, given the history of the word sequence already recognized, in order to create the acoustic hypotheses used for the successive pattern matching (HMM recognition) process. Considering (3), the first case (i.e. immediately after 'shift' states, S_s) is decomposed as follows.

$$P(l_i a_i | s_{i-1}) = P(l_i | s_{i-1}) P(a_i | l_i s_{i-1}). \tag{4}$$

Since the term $P(l_i|s_{i-1})$ represents the prediction of the next word, the history is s_{i-1}, the state immediately after the 'shift' action. The other terms of (3) (i.e. $P(a_i|l_i s_{i-1})$) can be considered that they predict the action, i.e. the structure of the resulting parse tree T.

Instead of using the states immediately after 'shift', states after 'reduce' action can also be used for predicting the next word, and sometimes they behave better for the history. There seems to be no reason not to use these states if they are effective, so from this point of view we have extended the original PGLR model.

2.2. Abstracted Probabilistic GLR model

In (1), the action a_i is ether 'shift' or 'reduce'. From the action sequence $a_1 \cdots a_n$, we can pick out the unique subsequence that consists of all the 'shift' actions $a_{x(1)} a_{x(2)} \cdots a_{x(m)}$, in which the order is reserved. In addition, we assume $x(0) = 0$ and $x(m) = n$. Considering the first action a_1 is 'shift', equation (1) can be rewritten as follows.

$$\begin{aligned} P(T) =& P(\sigma_0 l_{x(1)} a_{x(1)} \sigma_{x(1)} l_{x(1)+1} a_{x(1)+1} \sigma_{x(1)+1} \cdots \\ & \vdots \\ & \cdots l_{x(k)} a_{x(k)} \sigma_{x(k)} l_{x(k)+1} a_{x(k)+1} \sigma_{x(k)+1} \cdots \\ & \cdots l_{x(k+1)} a_{x(k+1)} \sigma_{x(k+1)} \cdots \\ & \vdots \\ & \cdots l_{x(m)} a_{x(m)} \sigma_{x(m)}) \end{aligned} \tag{5}$$

Between the interval from an immediately after 'shift' state to the next immediately before 'shift' state (i.e. $\sigma_i (x(k-1) < i \leq x(k))$, the lookahead symbols $l_i (x(k) < i \leq x(k+1))$ are same ($l_i = l_{x(k+1)}$ for $x(k) < i \leq x(k+1)$). Defining this interval as one portion of the parsing, equation (5) can be decomposed as follows.

$$\begin{aligned} &= P(\sigma_0) \prod_{k=1}^{m} P(l_{x(k-1)+1} a_{x(k-1)+1} \sigma_{x(k-1)+1} \cdots \\ &\qquad \cdots l_{x(k)} a_{x(k)} \sigma_{x(k)} \\ &\qquad | \ \sigma_0 \cdots l_{x(k-1)} a_{x(k-1)} \sigma_{x(k-1)}) \end{aligned} \tag{6}$$

$$\begin{aligned} &= P(\sigma_0) \prod_{k=1}^{m} P(l_{x(k)} a_{x(k-1)+1} \sigma_{x(k-1)+1} \cdots a_{x(k)} \sigma_{x(k)} \\ &\qquad | \ \sigma_0 \cdots l_{x(k-1)} a_{x(k-1)} \sigma_{x(k-1)}) \end{aligned} \tag{7}$$

$$\begin{aligned} &\approx P(\sigma_0) \prod_{k=1}^{m} P(l_{x(k)} a_{x(k-1)+1} \sigma_{x(k-1)+1} \cdots a_{x(k)} \sigma_{x(k)} \\ &\qquad | \ \sigma_{x(k-1)}) \end{aligned} \tag{8}$$

In equation (8), we assume that the parsing process can be predicted only by the immediately preceding parser's state.

Each term in the product of (8) that corresponds to step k can be decomposed as follows.

$$P(l_{x(k)} a_{x(k-1)+1} \sigma_{x(k-1)+1} \cdots a_{x(k)} \sigma_{x(k)} | \sigma_{x(k-1)}) \tag{9}$$

$$\begin{aligned} =& P(l_{x(k)} a_{x(k-1)+1} a_{x(k-1)+2} \cdots a_{x(k)} | \sigma_{x(k-1)}) \cdot \\ & P(\sigma_{x(k-1)+1} \cdots \sigma_{x(k)} | l_{x(k)} \sigma_{x(k-1)} a_{x(k-1)+1} \cdots a_{x(k)}) \end{aligned} \tag{10}$$

$$= P(l_{x(k)} a_{x(k-1)+1} a_{x(k-1)+2} \cdots a_{x(k)} | \sigma_{x(k-1)}) \tag{11}$$

Notice that the second term of (10) is always 1, because the sequence of the parser's states $\sigma_{x(k-1)+1} \cdots \sigma_{x(k)}$ is deterministically predicted, giving both the immediately preceding parser's state $\sigma_{x(k-1)}$ and the following sequence of the action $a_{x(k-1)+1} \cdots a_{x(k)}$ after $\sigma_{x(k-1)}$.

Now we set $y(k)$ that satisfies $x(k-1) \leq y(k) < x(k)$ for $k = 1 \cdots m$. Giving $y(k)$, equation (11) can be decomposed as follows.

$$\begin{aligned} =& P(a_{x(k-1)+1} \cdots a_{y(k)} | \sigma_{x(k-1)}) \\ & \cdot P(l_{x(k)} | \sigma_{x(k-1)} a_{x(k-1)+1} \cdots a_{y(k)}) \\ & \cdot P(a_{y(k)+1} \cdots a_{x(k)} | l_{x(k)} \sigma_{x(k-1)} a_{x(k-1)+1} \cdots \\ & \quad \cdots a_{y(k)}) \end{aligned} \tag{12}$$

$$\begin{aligned} =& \{ \prod_{i=x(k-1)+1}^{y(k)} P(a_i | \sigma_{x(k-1)} a_{x(k-1)+1} \cdots a_{i-1}) \} \\ & \cdot P(l_{x(k)} | \sigma_{x(k-1)} a_{x(k-1)+1} \cdots a_{y(k)}) \\ & \cdot \{ \prod_{j=y(k)+1}^{x(k)} P(a_j | l_{x(k)} \sigma_{x(k-1)} a_{x(k-1)+1} \cdots a_{j-1}) \} \end{aligned} \tag{13}$$

Giving both the parser's state $\sigma_{x(k-1)}$ and the following sequence of actions $a_{x(k-1)+1} \cdots a_i$, the following parser's state σ_i is deterministically predicted. Thus the conditions of each term in (13) approximate the preceding parser's state (stack), and furthermore, the state of the LR table appearing on the top of the stack.

$$\begin{aligned} \approx& \{ \prod_{i=x(k-1)+1}^{y(k)} P(a_i | s_{i-1}) \} \\ & \cdot P(l_{x(k)} | s_{y(k)}) \\ & \cdot \{ \prod_{j=y(k)+1}^{x(k)} P(a_j | l_{x(k)} s_{j-1}) \} \end{aligned} \tag{14}$$

Note that there are several variants of LR tables; the most commonly used variants are a Simple LR, a Canonical LR, and a Look-ahead LR (LALR)[7]. The above approximation is proper for a Simple LR and a Canonical LR. However, it is not proper for a LALR because some states of the LALR table are merged in spite of the different states of parsing process, and they predict inadequate lookahead symbols. In order to deal with LALR

[1] For convenience sake, the initial state is classified in S_s

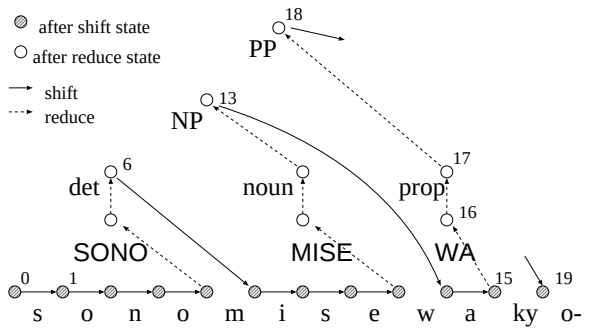

Figure 1: GLR parsing process

tables, the equation (13) approximates in the following way,

$$\begin{aligned}\approx\{&\prod_{i=x(k-1)+1}^{y(k)} P(a_i|L_{\sigma_{x(k-1)}a_{x(k-1)+1}\cdots a_{i-1}}s_{i-1})\}\\ &\cdot P(l_{x(k)}|L_{\sigma_{x(k-1)}a_{x(k-1)+1}\cdots a_{y(k)}}s_{y(k)})\\ &\cdot\{\prod_{j=y(k)+1}^{x(k)} P(a_j|l_{x(k)}s_{j-1})\}\end{aligned} \tag{15}$$

where $L_{\sigma_{x(k-1)}a_{x(k-1)+1}\cdots a_i}$ is the set of the lookahead symbols $l_{x(k)}$ that can do actions $a_{x(k-1)+1}\cdots a_i$ from parser's state $\sigma_{x(k-1)}$.

2.3. The Meaning of Abstracted PGLR Model

Figure 1 illustrates the process of GLR parsing using the LR table whose lookahead symbols are phonemes. Such a phoneme-based LR table can considerably suppress redundant search process by sharing the prefix of phoneme sequence with multiple words hypotheses. It can also allows us to treat allophones within the LR table[8].

The term $y(k)$ provides the abstraction level that is used for the prediction of the next word. In figure 1, suppose the parser already have processed the sequence "s o n o m i s e w a", i.e. at step 11 ($k = 11$). Since $x(10) = 15$ and $x(11) = 19$, $y(11)$ should be selected in $15 \leq y(11) < 19$. If $y(11)$ is selected as $y(11) = 15$, the next word is predicted from the state s_{15}, which is the LR state immediately after recognizing the phoneme 'a'. If $y(11) = 16$, the word is predicted from the s_{16}, which is the state immediately after recognizing the word 'WA', the subject marker in Japanese. If $y(11) = 17$, the word is predicted from s_{17}, which is the state after recognizing a postposition (`prop`). If $y(11) = 18$, then s_{18}, after a postpositional phrase (`PP`). [2]

The strategy for selecting $y(k)(k = 1\cdots m)$ varies. For example, we can set $y(k) = x(k) - 1$ for all k, which results in selecting the word prediction from the most abstracted grammatical category for each k. We also can choose the sort of categories that are used to predict the next word by means of selecting $y(k)$s that refer to the states of the LR table corresponding to the categories. Moreover, we can dynamically select $y(k)$s in the midst of the parsing process. Several examples of this selection will be shown in section 3.

[2]Note that the states of the LR table s_i are more informative than the corresponding grammatical categories, since they are separated by, i.e. have information about, the parsing process up to them.

Attributes	G_1	G_2
rules	1302	616
words	385	384
coverage (%)	62.8	65.2
word perplexity (without probability)	9.80	23.8
average number of trees per sentence	1.05	1.98

Table 1: The attributes of grammars

In particular, when we set $y(k) = x(k-1)(k = 1\cdots m)$, it results in selecting the word prediction from the least abstracted grammatical categories, which are terminal symbols, for each k. In this case, equation (14) can be rewritten as follows.

$$\begin{aligned}=&1\cdot P(l_{x(k)}|s_{x(k-1)})\cdot\prod_{j=x(k-1)+1}^{x(k)} P(a_j|l_{x(k)}s_{j-1})\\ =&P(l_{x(k)}a_{x(k-1)+1}|s_{x(k-1)})\\ &\cdot\prod_{j=x(k-1)+2}^{x(k)} P(a_j|l_{x(k)}s_{j-1})\end{aligned} \tag{16}$$

This is equivalent to the original PGLR model (equation (3)), because the first term corresponds to the immediately after 'shift' state($s \in S_s$) and the other terms correspond to the immediately after 'reduce' states ($s \in S_r$). This indicates that the APGLR model is a generalization of the original PGLR model.

One of the advantages the APGLR model holds over the PGLR model is its ability to cope with the data sparseness problem. The more a category is abstracted, the more samples for statistics it tends to have. The APGLR model can be used to smooth the probability by increasing the abstraction level. Moreover, it is said that the local word history works well in a phrase, but is unsuccessful at a phrase boundary. The APGLR model can flexibly change the abstraction level of the history, i.e., the less abstracted categories in phrases, and at the same time, the more abstracted categories at phrase boundaries.

3. Experimental Results

We have implemented the APGLR model mentioned in section 2.2 into the existing speech recognition system based on LR parsing algorithm[6]. The system is the one-pass decoder that considers both the language model score and the acoustic model score at the same time; the weighted score from the language model is added to the acoustic score from HMMs at each end of terminal symbols (in our case, phonemes). The integrated score is considered for beam search at every frame.

The two grammars for a town guidance task[9] were used in the following experiments. The attributes of the grammars are shown in table 1. For each grammar, the LR table that has the terminal symbols corresponding to phonemes was constructed.

The user utterances (about 4000 words) of the town guidance corpus[9] were used to train our statistical models. We determined the abstraction level (i.e. $y(k)$ shown in section 2.2) in several ways and obtained several APGLR models.

The model labeled '$A(l)$-PGLR' is defined as follows:

$$y(k) = \begin{cases} x(k-1)+l & \cdots x(k-1)+l < x(k) \\ x(k)-1 & \cdots x(k-1)+l \geq x(k) \end{cases} \tag{17}$$

Note that '$A(0)$-PGLR' is equivalent to the original PGLR model. According to our design of grammars, phonemes,

model	G_1	G_2
$A(0)$	7.113	9.012
$A(1)$	6.961	9.029
$A(2)$	6.974	9.517
$A(3)$	7.017	10.04
$A(\infty)$	7.015	10.42
A(flex)	7.031	10.44
2-gram	8.736	10.21
3-gram	8.167	9.503

Table 2: Test set perplexity

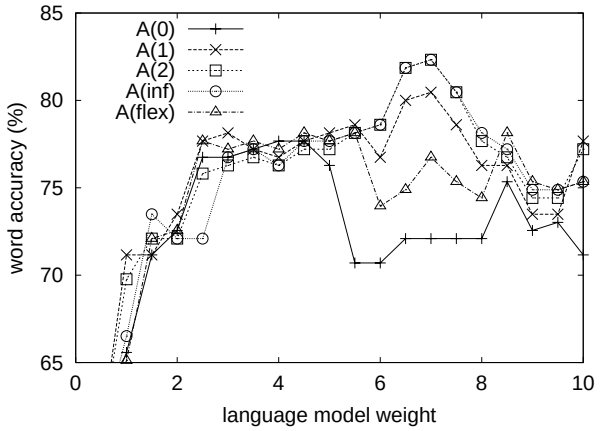

Figure 2: Word accuracy using G_1

words and part of speech are used to predict the next word in the '$A(0)$-PGLR', '$A(1)$-PGLR', '$A(2)$-PGLR' models respectively.

The model labeled '$A(\infty)$-PGLR', which selects most abstracted categories at any position, is defined as follows:

$$y(k) = x(k) - 1 \tag{18}$$

In the midst of parsing process, the model labeled 'A(flex)-PGLR' dynamically, selects the level ($x(k-1) \leq y(k) < x(k)$) that has most reliable statistics, i.e., the one having the largest samples.

A test set (623 words for G_1 and 698 for G_2 according to the difference on the coverage) was selected apart from the training set from the corpus. The test set perplexity was calculated from them (table 2). The word accuracy of them, obtained through recognition experiments on a part of the test set (215 words), were also investigated (figure 2,3). All of the newly proposed APGLR models improve the perplexity on the grammar G_1, but degrade on G_2, although the differences are slight. On the other hand, the word accuracy are improved by the proposed models on both G_1 and G_2. On the whole, the abstracted models with a fixed level, such as the $A(2)$-PGLR models, performed better among our models.

4. Conclusion

A new statistical language model, the Abstracted Probabilistic GLR model, is proposed. The APGLR model is both an extension and a generalization of the Probabilistic GLR model. The formalization of the APGLR model is presented. We informally showed how it is expected to work well. The model can

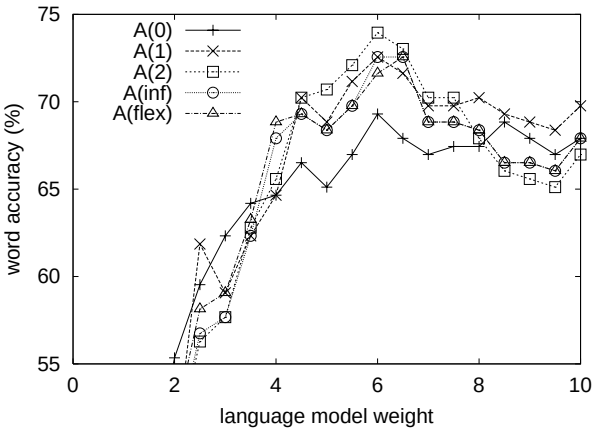

Figure 3: Word accuracy using G_2

adopt an arbitrary level of abstracted categories at every next word prediction, and we showed several strategies for setting this level and defined several variations of the APGLR model. The experimental results showed the improvement on the word accuracy compared with the original PGLR model.

5. References

[1] E. Charniak and G. Carroll. Context-sensitive statistics for improved grammatical language models. Proceedings of AAAI-94, pp.728-733, 1994.

[2] C. Chelba and F. Jelinek. Exploiting syntactic structure for language modeling. In Proceedings for COLING-ACL 98, pp.225-231, 1998.

[3] M. Tomita. An Efficient Parsing for Natural Language: A Fast Algorithm for Practical Systems. Kluwer Academic Publishers, Boston, Mass.

[4] T. Briscoe and J. Carroll. Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars. Computational Linguistics, Vol. 19, No. 1, 1993.

[5] K. Inui, V. Sornlertlamvanich, H. Tanaka, and T. Tokunaga. A new formalization of probabilistic GLR parsing. In Proceedings of the 5th International Workshop on Parsing Technologies, pp. 123-134, 1997.

[6] 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. 1992 International Conference on Acoustics, Speech and Signal Processing, pp. I-21-24, 1992.

[7] A. V. Aho et al. Compilers: Principles, Techniques, and Tools. Addison Wesley, 1986.

[8] H. Li and H. Tanaka. A method for integrating the connection constraints into an LR table. In Proceedings of the Natural Language Pacific Rim Symposium (NLPRS95), pp.703-708, 1995.

[9] K. Itou, T. Akiba, O. Hasegawa, S. Hayamizu, and K. Tanaka. A Japanese spontaneous speech corpus collected using automatically inferencing Wizard of OZ system. The Journal of the Acoustical Society of Japan, Vol.20, No.3, May 1999.