2007

The Use of Context-free Grammars in Isolated Word Recognition

Chaiyaporn Chirathamjaree
Edith Cowan University

10.1109/TENCON.2004.1414551
This is an Author's Accepted Manuscript of: Chirathamjaree, C. (2004). The use of context-free grammars in isolated word recognition. Proceedings of TENCON 2004 . (pp. 140-143). Chiang Mai, Thailand. IEEE Thailand Section. Available here

THE USE OF CONTEXT-FREE GRAMMARS IN ISOLATED WORD RECOGNITION

Chaiyaporn Chirathamjaree

Edith Cowan University, Australia

ABSTRACT

A method using non-recursive context-free grammars is presented for the recognition of isolated words. Some form of 'training' is required to combat problems of variations in speech. In the training mode, one grammar for each word in the vocabulary is constructed directly from a set of sample strings of 'features' represented by symbols. In the recognition mode, an incoming string is analyzed to determine which grammar, if any, could have generated it. The word corresponding to such grammar is then said to have been recognized.

1. INTRODUCTION

An isolated word recognition (IWR) system is one which can recognize human utterances. Short pauses are required before and after utterances to be recognized. Following the common practice in the filed of pattern recognition [1], an IWR system can be considered to consist of a feature extractor (FE) or a pre-processor of some sort followed by a recognizer or classifier (Figure 1). The FE transcribes the input speech signal into strings of symbols representing various parameters extracted from the signal. A decision is made by the recognizer on this simplified representation as to which word in the vocabulary, if any, has been spoken.

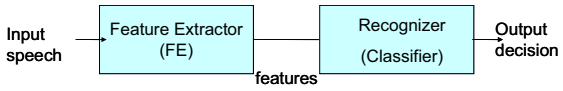

Figure 1: An Isolated-Word Recognition (IWR) System

The classical decision-theoretic methods [1,2] from the filed of pattern recognition have commonly been used to produce recognizers for processing strings of symbols generated by the FE. Another promising approach, which stemmed from the fields of mathematical linguistics and computer science, is to make use of the techniques of formal language theory [3]. The essence of this method is to classify a pattern by determining which of a number of *formal grammars* [4] or sets of rules could have generated it. This paper presents the application of linguistic methods to the design and implementation of recognizers of isolated words from a limited vocabulary.

0-7803-8560-8/04/$20.00©2004IEEE

2. THE CFG IN WORD RECOGNITION

In general, a person does not always speak the same word in the same way. This may be due to the emotional and physical states of the speaker, the ambient noise level of the surrounding and free variation from trial to trail. Hence, some form of 'training' is usually required in order to combat problems of variations of speech. This is done by having the user speaking each word in the vocabulary a number of times until the representative rules for the construction of each word are formed.

In IWR systems, words are spoken in isolation with short gaps between utterances. This leads to the following assumptions:-

(a) Only a finite number of symbols are generated by the FE and only one symbol can be represented at a particular time.

(b) Each word uttered results in a sequence of symbols of some finite length.

The problem of designing a recognizer in an IWR system can be broadly divided into two areas: the construction of models based on formal grammars, known as grammar inference [5,6,7], to represent characteristics of the symbol-generating source and the search for suitable decoding methods for efficiently analysing the strings from the source using rules or grammars of the models previously created.

In outline, the basis of the linguistic method is simply explained (Figure 2). In the training (learning) mode, a user repeats each word in the vocabulary a number of times. Each time the same word is spoken, a similar but not necessarily the same string of symbols is produced by the FE. Grammar-based models, one for each word in the vocabulary, are then automatically constructed and stored in the system memory for future use. In addition to producing all the strings in the sample set, each model is also capable of predicting other similar strings. Model building is considered to be the encoding of strings. In the recognition mode, an incoming string is processed using suitable decoding algorithms to determine which model, if any, corresponds or nearly so to the word spoken. If the most compatible model is found, the corresponding word is then indicted as to have been recognized. Otherwise, the recognition fails and the word is rejected.

In IWR systems, unlike many applications of grammar inference where the class of grammars to be inferred is precisely defined, it is not clear what types of grammars best represent the FE. This paper

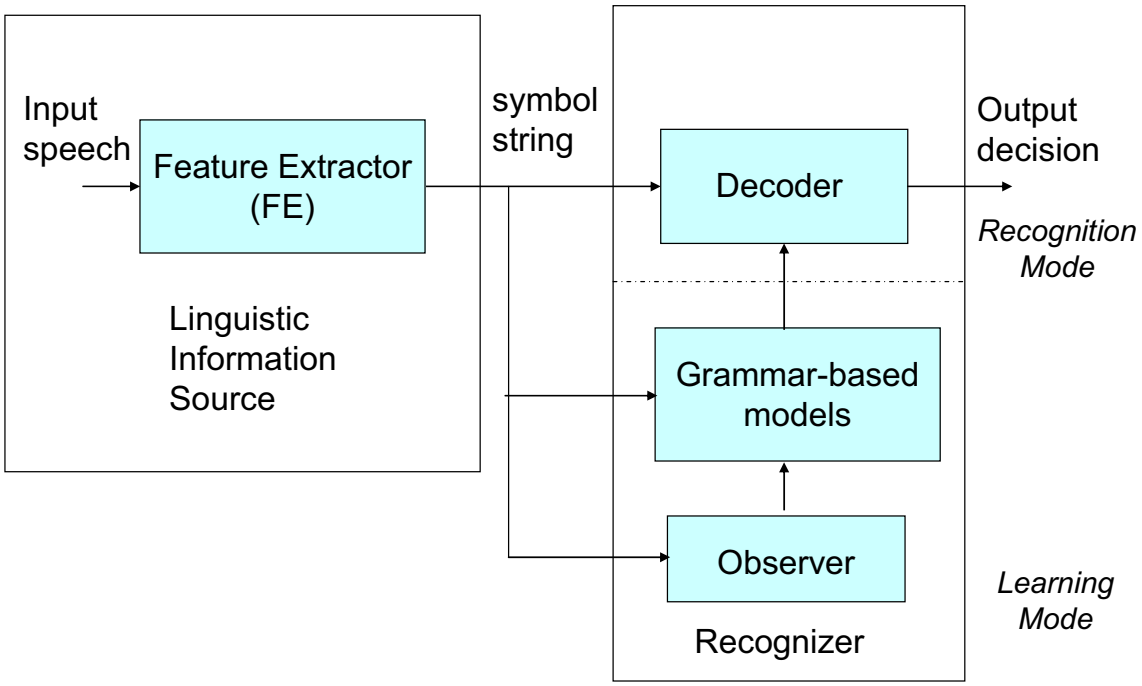

Figure 2: A Grammar-Based IWR System

presents a method using non-recursive context-free grammars (CFGs) for the recognition of isolated words. This would be adequate since each training set can consist only of a finite number of finite-length strings. The CFG approach is selected because it enables useful complexity to be generated without requiring inordinate computing power. In addition, a CFG may have fewer nodes and/or links than a finite-state grammar using the same data.

Some notation and terminology are now introduced. Precise and complete definition of context-free grammars (CFGs) are available elsewhere [3,4,8]. A CFG consists of:

1. A finite set of *non-terminals* $A_1, A_2, \ldots, A_r$;
2. A finite set of *terminals* $b_1, b_2, \ldots, b_s$;
3. A set of *rewriting rules* (or just *rules*) of the form $A \rightarrow \beta$; where A is a non-terminal and β is a non-empty string of terminals or non-terminals, or both;
4. A *start symbol*, which is one of the non-terminals.

Any CFG can be transformed into an equivalent Chomsky normal form [4] in which the rules are of the following forms only:

Bielement rules: $A \rightarrow BC$

Terminating rules: $A \rightarrow a$

where A, B and C are non-terminals and a is a terminal.

3. COMPUTATION OF THE MINIMIZATION MATRIX

Before the inference method can be given, it is necessary to describe the *minimization matrix*, M, which forms the basis of the inference process.

For a given CFG in Chomsky normal form and for a string, S, the minimization matrix, M, (M-matrix) is a 3-dimensional, $n \times n \times r$ matrix where n is the length of S and r is the number of non-terminals in the grammar. Let a_j represent the j^{th} symbol of S. Element m_{ijk} of M denotes the minimum number of symbol alterations (insertions, deletions or substitutions) required if the length-i substring of S, whose first symbol is a_j, is to be generated by the grammar from A_k, the k^{th} non-terminal.

M-matrix can be computed iteratively by the following procedure.

Part 1: Terminating rules

$m_{1jk} = 0$, if and only if $A_k \rightarrow a_j$ is a rule of the CFG

$m_{1jk} = 1$, otherwise.

For i=2,3,…,n:

$$m_{ijk} = \max\left[(i-1), \sum_{u=j}^{j+i-1} m_{1uk} \right].$$

Part 2:Bielement rules

$$m_{1jk} = \min_{p,q \in P_k} \{\min[m_{1jp} + H(q), m_{1jq} + H(p)]\}$$

where P_k is the set of ordered pairs (p,q) such that $A_k \rightarrow A_pA_q$

For i=2,3,…,n:

$$m_{ijk} = \min_{p,q \in Pk} \left\{\min\left[x, \min_{1 \le u < i} (m_{ujp} + m_{i-u,j+u,q})\right]\right\}$$

where $x = [m_{ijp} + H(q), m_{ijq} + H(p)]$

4. CFG INFERENCE PROCEDURE

In the learning mode, a learning algorithm is employed to automatically construct CFGs, one for each word in the specified vocabulary. The inference process produces rewriting rules directly from the observed sample strings in response to the words spoken. Rule or production probabilities are also estimated during the learning process.

Inference algorithms are based on the criterion of maximising the similarity between various strings of the same word. The basis of the inference process is now explained. The first grammar, called the *skeleton grammar* G_1, is constructed from the first string in the sample set such that G_1 can generate only that string. Other strings are then individually processed in the search for incompatibility between each string and the current grammar. If the nth observed string, S_n can be derived from the $(n-1)^{th}$ inferred grammar G_{n-1}, then $G_n = G_{n-1}$ and no augmentation of G_{n-1} is required. Otherwise, G_{n-1} is augmented such that G_n is produced which can generate the present string. The matching process involves the computation of the M-matrix whose elements reveal the shortcomings of the CFG in relation to its ability to generate the string.

5. RECOGNITION SCHEME

In the recognition mode, each unknown string presented to the recognizer is classified or decoded using the rules obtained earlier. The recognition process consists of three main levels of operation in terms of the complexity involved. The recognition always starts at the lowest level (level 1). A higher level is applied only if the previous one fails to classify a string according to some criteria.

Level 1: In this simplest level of the recognition process, an incoming string is tested by means of a parsing algorithm to determine which grammar, if any, could have generated it. If the string is accepted by one grammar only, the corresponding word is indicated at the output. For unsuccessful matching, the method of level 2 is applied to decode the string. When two or more grammars can generate the string, it is necessary to employ a stochastic algorithm to find one 'best word' that is most likely to have produced the string. If two or more such words are possible, the string is rejected.

Level 2: In this level, a technique is utilized to determine the 'closest match' for the string i.e. the grammar that could nearly have generated the string. It is basically a dynamic programming method of optimizing the similarity between two functions.

Level 3: This is the highest and most complicated level of the recognition process where the operations in the two lower levels have to be performed in order to reach level 3. It is applied when there exist two or more closest-matched words corresponding to the string. Another stochastic algorithm is employed to select the most likely closest-matched word. The string is rejected if two or more such words are found.

The foregoing recognition scheme is not too restrictive in the sense of immediate rejection of an erroneous string but rather trying to find a grammar that could most likely have generated the string. This can be very useful in many applications involving noisy strings.

6. RESULTS AND DISCUSSION

The experimental IWR system has been constructed based on the use of the CFGs outlined in this paper to model the FE. The vocabulary consists of ten digits 'ZERO' to 'NINE' uttered by a single speaker. The speech signal of the spoken digits is of telephone-grade quality. This is obtained from a normal telephone set via a circuit representing two limiting local lines. Ten CFGs for the recognition of the spoken digits 'ZERO' to 'NINE' were generated from a total of 100 strings, of average length 3.6 symbols. Table 1 shows the complexity measure of the inferred CFGs.

The previous sections have described an incremental method for the construction of non-recursive CFGs. This is appropriate for applications such as the recognition of isolated words, where finite-length strings only are involved. The method presented is guaranteed to yield CFGs that are capable of producing all the given strings, irrespective of the order in which they are presented. Any other strings generated by the grammar will resemble those in the training set. The method also produces compact CFGs having a near-minimal number of rules and non-terminals.

The representation of strings by a set of rules of formal grammars instead of direct storage of strings make possible the 'generalisation' of strings in the training set. This reduces the size of the training set needed compared with the approach of using template matching techniques in order to cover the same number of strings. Extension of the method to the construction of stochastic CFGs, by counting the frequency of use of rules, is straightforward. Future work on the recognition of words in connected speech is planned.

7. CONCLUSION

Techniques of formal language theory or the syntactic methods provide a useful approach to the solution of classification and description in an isolated word recognition system where only a finite number of features (represented by symbols) are generated for each utterance. The method presented for the generation of CFGs from sample strings enables grammars of useful complexity to be generated without requiring inordinate computing power.

8. REFERENCES

[1] P. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley, New York, 2001.

[2] K. S. Fu, *Syntactic Pattern Recognition and Applications*, Prentice-Hall, Englewoods Cliffs, 1982.

[3] N. Chomsky, *Aspects of the Theory of Syntax*, MIT Press, Cambridge, Massachusetts, 1964.

[4] J. E. Hopcroft, R. Motwani, and J. D. Ullman, *Introduction to Automata Theory, Languages, and Computation*, 2nd Ed., Addison-Wesley, Reading, Massachusetts, 2000.

[5] K. Clarkson, and T. G. Dietterich, *Grammatical Inference, The Handbook of Artificial Intelligence*, pp. 494-511, William Kaufmann, Inc, 1982.

[6] T. G. Evans, "Grammatical inference techniques in pattern analysis", *Software engineering*, Volume (2), pp. 183-202, Academic Press, 1971.

[7] L. Miclet, *Grammatical Inference, Syntactic and Structural Pattern Recognition: Theory and Applications*, World Scientific, 1990.

[8] A. V. Aho, and J. D. Ullman, *The Theory of Parsing, Translation, and Compiling, Vol. I: Parsing*, Prentice-Hall, Englewood Cliffs, New Jersey, 1972.

Words spoken	ONE	TWO	THREE	FOUR	FIVE	SIX	SEVEN	EIGHT	NINE	ZERO
No. of terminals	13	10	11	19	27	18	18	21	12	15
No. of non-terminals	4	3	3	14	29	28	18	21	3	31
No. of rules	24	22	21	42	65	57	45	53	23	59

Table 1: Complexity Measure of Inferred CFGs