Grammar Generation and Optimization from Multiple Inputs

Pankaj B. Devre,
Department of Computer Engg,
MET's Institute of Engg., Nashik,
Savitribai Phule Pune University,
pbdevre@gmail.com

Prof. Madhuri A. Bhalekar,
Department of Computer Engg.
MAEERS MIT Pune,
Savitribai Phule Pune University,
madhuri.chatur@gmail.com

Dr. Madan U. Kharat
Department of Computer Engg,
MET's Institute of Engg., Nashik,
Savitribai Phule Pune University,
madank_iot@bkc.met.edu

Abstract— Human being uses multiple modes like speech, text, facial expression, hand gesture, showing picture etc. for communication in between them. The use of this ways for communication makes human communication more simple and fast. In previous years several techniques are used to bring the human computer interaction more closely. It costs more for development and maintenance of Multimodal grammar in integrating and understanding input in multimodal interfaces i.e. using multiple input ways. This leads to improve and investigate more robust algorithm. The proposed system generates the grammar from multiple inputs called as multimodal grammar and evaluates grammar description length. Furthermore, to optimize the multimodal grammar proposed system uses learning operators which improves grammar description.

Index Terms— *Context-Free-Grammar, Grammar Description length, learning operators, Multimodal Grammar.*

*****

I. INTRODUCTION

Interaction between the human being is naturally by using multiple ways such as facial expressions, speech, handwriting, gestures, drawing etc. or combination of such modes. Multimodal interfaces that allow input and/or output to be conveyed over multiple channels enable more natural and effective mode of interaction. Here we want to apply same paradigm in between Human-Computer interaction to close the human-computer communication. Therefore the multimodal interfaces, which have been used to communicate with computer through several input channels, have gained more importance in research field. Multimodal grammar provides a methodology [1]–[3][7-8] in integration of inputs in several multiple input interfaces. In this thesis, the outcomes of each unimodal recognizer are considered as terminal symbols of the grammar, and they are recognized by the parser as a unique multimodal sentence. Therefore, in the interpretation phase, the parser uses the production rules to interpret each multimodal sentence.

However, for writing and maintaining a grammar, it needs a highly skilled person. For defining the grammar, a skilled grammar developer has to build a body of multimodal sentences and has to generate by hand the initial grammar. After the grammar is deployed into system, the necessity of new multimodal sentences arises and the grammar developer has to manually update the grammar in order to include also these sentences. This manual process is high costing and time consuming.

Hence a way to overcome these difficulties is to automate the grammar generation and updation. Generally, a grammar inference algorithm works by taking in as input a finite sequence of examples and by giving out as output the grammar rules that are able to derive these examples.

To generate the multimodal grammar, an efficient algorithm for grammatical inference, which extends the CYK algorithm proposed by Cocke-Younger-Kasami [4] for integrating multimodal sentences, has been developed. This algorithm allow us to learn the multimodal grammar from positive sample of strings. And to avoid over generalization problem an e-GRIDS algorithm [6] is used which makes use of learning operators and minimum description length metrics.

II. LITERATURE SURVEY

The theory of language learning related with the procedures in acquiring grammars from the languages. The main research studies in generating grammar have been done in several application domains, such as speech recognition, computational linguistics, computational biology, and machine learning. Most of these learning models take as input an set of training examples and as outputs the grammar i.e. language description. This generated grammar is able to parse the sentences from which it is generated. Majority of natural language grammar inference algorithms focus on generating context-free grammars [9] (CFGs). This kind of grammar is similar to Context Free Grammar. The context-sensitive Grammar in which expansion of a symbol depends on its position i.e. on context of the symbol. Context-sensitive grammars are well known algorithms for parsing the grammars that have exponential time dependency.

Here, context-sensitive have more expressive power than CFGs i.e. they are able to model all frequent linguistic

ISSN: 2321-8169
298 - 303

phenomena of natural language and multimodal language at the same time and with lower parsing complexity.

The examples of existing algorithms for grammatical inference of CFGs are as follows: the *inductive CYK algorithm* [4], the *learning by version space algorithm* [5], and the *e-GRIDS algorithm* [6].

The *inductive CYK algorithm* is implemented by analyzing Positive samples of examples (Synapse) [4]. The algorithm synthesizes CFGs from positive and negative sample of strings generating the production rules, which derives positive strings of samples but do not derive any given negative strings of samples. All of the production rules outputted by the algorithm follows Chomsky normal form, such as $A \rightarrow \beta$ and $A \rightarrow BC$, where *A*, *B*, and *C* are nonterminal symbols and β is a terminal symbol. The main advantages of the extended inductive CYK algorithm is the generation of simple sets of production rules and shorter computational time compared to the other grammatical inference algorithms. A limitation of the grammar inference method is that the computation time is dependent on the length of the positive sample strings, and it becomes inefficient when the longer positive samples are given first.

The *learning by version space algorithm* [5] uses both the positive and negative examples. A version space is a set of all generalizations of a grammar, which is consistent with a given set of instances. The algorithm applies a version space strategy, which is based on method of compact way of representing the version space. In this the space of generalizations can be partially ordered and application of this criterion produces a space of generalizations which covers only a few sentences. It uses three operators Update, Done and classifies to search for the appropriate set of generalizations. The drawback of this algorithm is that it is not immediately applicable to grammar because it produces a set of grammars and other processes needed to select among them. Hence it can be used for task-specific learning machines.

The *e-GRIDS algorithm* [6] is a grammar inference method that uses positive samples of training sentences to construct an initial grammar. The learning process, which having an initial grammar, e-GRIDS uses three learning operators in order to explore the space of CFGs: the *MergeNT*, *CreateNT*, and *Create Optional NT* operators. One of the main advantages of the e-GRIDS algorithm is its computational efficiency and scalability to large example sets.

The grammar inference algorithm proposed in this thesis joins together the strengths of the inductive CYK and e-GRIDS algorithms, adapting them to multimodal input. The proposed method learns from positive examples because it is problematic to consider negative examples due to high potentially infinite numbers.

The grammar inference algorithm in this thesis combines together the strength areas of inductive CYK and e-GRIDS algorithm. The disadvantage of the CYK algorithm relies on the computation time taken by the algorithm for the positive sample of sentences. The required time will increases if the positive sample of sentences are longer and it creates over generalization problem

The e-GRIDS algorithm uses the two learning operators create and merge which merely optimize the grammar by combining the longest common subsequence into the unique subsequence. So, more operators are needed in order to maximize the grammar description length and avoiding remaining unreachable states which will not be used by the grammar afterwards.

Moreover, the multiple input channels which has multiple modes has to be need to first generate the natural language grammar. Afterwards by using updated Stanford parser our system generates the natural language grammar which will later combined to form the initial multimodal language grammar. Thereafter, CYK algorithm is basically meant for to generate the natural language grammar only. So, adaptation of CYK algorithm is needed to form the multimodal grammar and its candidate production rules and set of semantic functions associated with that rules.

The optimization of multimodal grammar and problem of over generalization of grammar need to be solve by using learning operators such as create and merge which uses longest common subsequence algorithm and one more operator needed here, to avoid the unreachable states that have no use in the generated multimodal grammar and by avoiding them system improves the initial grammar description.

III. Multimodal Grammar Representation

In the proposed multimodal grammar algorithm, multimodal attribute grammars (MAGs) are used,

A MAG is a triple $G = (G, A, R)$

Where,

G CFG (T,N,P,S), with *T* as a set of terminal symbols, *N* as a set of nonterminal symbols, *P* as a set of production rules of the form.

$X_0 \rightarrow X_1\, X_2 \, . \, . \, . \, Xn$

Where,

$n \geq 1$, $X_0 \in N$ and $X_k \in N \cup T$ for $1 \leq k \leq n$ and $S \in N$ as a start symbol (or axiom);

ISSN: 2321-8169
298 - 302

A collection $(A(X))_X \in N \cup T$ of the attributes of the nonterminal and terminal symbols, such that, for each $X \in N \cup T$,
$A(X)$ is split in two finite disjoint subsets, namely, $I(X)$ (the set of inherited attributes of X) and $S(X)$ (the set of synthesized attributes). The set $S(X)$, with $X \in T$, includes a set of attributes $MS(X)$, called as a set of multimodal synthesized attributes, composed of the following four attributes:

$MS(X) = \{val, mod, synrole, coop\}$;

R collection $(Rp)p \in P$ of semantic functions (or rules).

1) val that expresses the current value (concept) of the terminal symbol. The domain of the attribute is the set of terminal symbols: $D_{val} = T$.
2) mod that represents the modality. The domain of the attribute is the set of modalities.
   $D_{mod} = \{speech, handwriting, gesture, sketch\}$.
3) synrole that conveys information about the syntactic role. The domain of the attribute is $D_{synrole} = \{$*noun phrase, verb phrase, determiner, verb, noun, adjective, preposition, deictic, conjunction*$\}$.
4) coop that expresses the modality cooperation type with other terminal symbols. The domain of the attribute is $D_{coop} = \{complementary, redundant\}$.

IV. ALGORITHMIC STRATEGY

The proposed system inputs the sentences from text and from list mode. This section describes steps taken by algorithms to complete the system. The inputs capture by user interface is linearized and then forwarded to the first phase of the algorithm i.e. to the inductive CYK algorithm. This algorithm is provided for generating the initial grammar from positive sample sentences, while the e-GRIDS learning operators are taken as starting point for improving the initial grammar description. The choice of the CYK algorithm has been opted by its simplicity and efficiency, while the e-GRIDS enables to improve the grammar description length and making it more accurate. Therefore, the proposed grammar inference method tries to join together the strengths of the inductive CYK and e-GRIDS algorithms, adapting them to multimodal input. In particular, this method consists of two main steps as shown in figure 1. The first step includes the inductive CYK algorithm for generating the multimodal attribute grammar that is able to parse the input sentence; the second step makes use of the e-GRIDS operators for improving the grammar description coming from the first step.

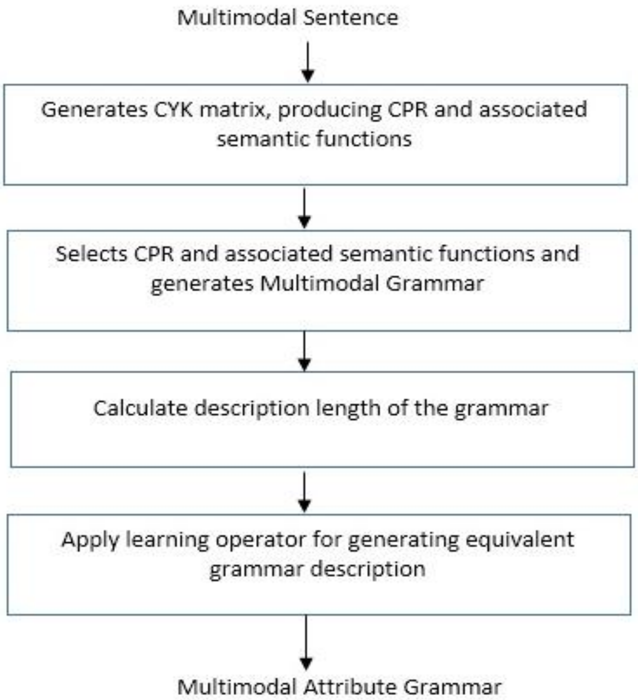

Fig. 1 Working Flow of the system

First Phase of Multimodal Grammar Algorithm:

The First step of the MGI algorithm enhances the inductive CYK algorithm in generating the MAG on two main aspects:

Input: An input sentence x : x_1, x_2.

Output: A CYK matrix C; a set CPR of candidate production rules.

Procedure:

1. Consider x as the sentence x_1, x_2
   Generate the set P' of production rules that is composed of rules of the form A_i -> x_i
2. Iterate the following processes for all $1 \leq i \leq k$
   i) Initialize a new CYK matrix C(k X k) by
   ii) Assign a weight
   iii) Assign to each c_{ij} a set of semantic functions.
2. Iterate the following processes for all $2 \leq j \leq k$ and $1 \leq i \leq k-j+1$
   i) Initialize the element $C_{ij} = 0$
   ii) For all $q(1 \leq q \leq j - 1)$
4. If $S \leq C_{ik}$ then return (success)
   Else proceed with step 2

This phase creates CYK Matrix qnd and set of candidate production rule as shown below.

CYK MATRIX

Start Of row:1
NN -> call (Weight: 0.5)(NN.val<-call)(NN.mod<-text)
IN -> that (Weight: 0.5)(IN.val<-that)(IN.mod<-text)|
NN1 -> company
(Weight:0.5)(NN1.val<-company)(NN1.mod<-text)|

NNS -> infosys
(Weight:0.5)(NNS.val<-infosys)(NNS.mod<-list)|
End Of row:1
Start Of row:2
B -> NN IN (Weight: 1.0)(B.val<-call)(B.mod<-text)|
C -> IN NN1 (Weight: 1.0)(C.val<-that)(C.mod<-text)|
D -> NN1 NNS (Weight:1.0)(D.val<-infosys)(D.mod<-list)|
End Of row:2
Start Of row:3
E -> NN C (Weight: 1.5)(E.val<-call)|
G -> IN D (Weight: 1.5)(G.val<-infosys)|
F -> B NN1 (Weight: 1.5)(F.val<-call)|
H -> C NNS (Weight: 1.5)(H.val<-infosys)|
End Of row:3
Start Of row:4
I -> NN G (Weight: 2.0)(I.val<-call infosys)|
J -> B D (Weight: 2.0)(J.val<-call infosys)|
K -> E NNS (Weight: 2.0)(K.val<-call infosys)|
End Of row:4
*****END of Matrix *****

Set of Candidate Production Rule

NN -> call (Weight:0.5)(NN.val<-call)(NN.mod<-text)
IN -> that (Weight: 0.5)(IN.val<-that)(IN.mod<-text)
NN1 -> company
(Weight:0.5)(NN1.val<-company)(NN1.mod<-text)
NNS -> infosys
(Weight:0.5)(NNS.val<-infosys)(NNS.mod<-list)
B -> NN IN (Weight: 1.0)(B.val<-call)(B.mod<-text)
C -> IN NN1 (Weight: 1.0)(C.val<-that)(C.mod<-text)
D -> NN1 NNS (Weight: 1.0)(D.val<-infosys)(D.mod<-list)
E -> NN C (Weight: 1.5)(E.val<-call)
G -> IN D (Weight: 1.5)(G.val<-infosys)
F -> B NN1 (Weight: 1.5)(F.val<-call)
H -> C NNS (Weight: 1.5)(H.val<-infosys)
I -> NN G (Weight: 2.0)(I.val<-call infosys)
J -> B D (Weight: 2.0)(J.val<-call infosys)
K -> E NNS (Weight: 2.0)(K.val<-call infosys)

Second Phase of Multimodal Grammar Algorithm:

During the second phase, e-GRIDS algorithm uses generated CYK matrix and CPR. It evaluates description length of the grammar and optimize multimodal grammar description.

Algorithm

Input: A CYK matrix C;
A set CPR of candidate production rules; a current multimodal attribute grammar *G ={G,A,R} with G = (T’,N’, P’,S’)*

Output: Description Length of the Grammar,
Optimized Multimodal attribute grammar

Procedure:

1. Select the non-terminal symbol A with the highest weight in the location c_{in} of the CYK matrix.
2. Find the candidate production rule r ∈CPR of the form r : A -> BC, containing A in the head, and consider the symbols B and C in the body.
3. Initialize *P’* <- P_0
4. Add the production *rules t: S->BC* to the set *P’*
5. Add the production *rule t : S -> BC* to the set *P’*
Else proceed with step 2
6. Iterate the following processes for all symbols in the body of a production rule: If B(C) is contained in the head of any rule of CPR.
7. Evaluate the description length DL of *G’*
8. Iterate the following processes
    a. For each production p ∈P
    b. Evaluate DL of the new grammar *G’’*

Following is the Multimodal Grammar Attribute generated by phase 1 of the algorithm along with set of start symbol, terminal symbol and non-terminal symbol.

Set of multimodal Grammar Attributes

```
S -> NN G (S.val<-call infosys)
NN -> call (NN.val<-call)(NN.mod<-text)
G -> IN D (G.val<-infosys)
IN -> that (IN.val<-that)(IN.mod<-text)
D -> NN1 NNS (D.val<-infosys)
NN1 -> company
(NN1.val<-company)(NN1.mod<-text)
NNS -> infosys
(NNS.val<-infosys)(NNS.mod<-list)
```

The second phase of the algorithm uses e-GRIDS algorithm for evaluating description length and applies learning operator to optimize the grammar description.
Following the approach proposed in [6], given a CFG G and a set of positive examples E, the description length DL of G is the sum of two independent lengths

DL = GDL + DDL

Where,

GDL, Grammar description length, i.e., the bits required to encode the grammar rules and to transmit them to a recipient who has minimal knowledge of the grammar representation;
DDL, derivation description length, i.e., the bits required to encode and transmit all examples in set E, provided that the recipient already knows the grammar G.

GDL and DDL Calculations

non terminal subset calculations result=
41.266194298518435

terminal subset calculations result= 99.86777022797764
start symbols subset calculations result= 22.51713206777987
Result After GDL Calculation= 163.65109659427594

Result of DDL calculations: 0.0

Grammar Optimization for candidate production rule using delete operator

NN -> call (Weight: 0.5)(NN.val<-call)(NN.mod<-text)
IN -> that (Weight: 0.5)(IN.val<-that)(IN.mod<-text)
NN1 -> company
(Weight:0.5)(NN1.val<-company)(NN1.mod<-text)
NNS -> infosys
(Weight:0.5)(NNS.val<-infosys)(NNS.mod<-list)
B -> NN IN (Weight: 1.0)(B.val<-call)(B.mod<-text)
C -> IN NN1 (Weight: 1.0)(C.val<-that)(C.mod<-text)
D -> NN1 NNS (Weight: 1.0)(D.val<-infosys)
E -> NN C (Weight: 1.5)(E.val<-call)
G -> IN D (Weight: 1.5)(G.val<-infosys)
S -> NN G (Weight: 2.0)(I.val<-call infosys)
J -> B D (Weight: 2.0)(J.val<-call infosys)
K -> E NNS (Weight: 2.0)(K.val<-call infosys)

V. Conclusion and Future Work

Multimodal interaction in between human and computer has emerging in the last few years as the future paradigm of human-computer interaction. Multimodal communication requires that several simultaneous inputs, coming from various input modalities, are integrated and combined into a complete sentence, this paper represents an approach of grammar definition that follows the training sentences paradigm, that is, the language designer provides concrete examples of multimodal sentences that have to be recognized by the system through different modalities and , and a multimodal grammar algorithm that automatically generates the grammar rules to parse those examples. Also, by applying different learning operators system improves grammar description for grammar optimization.

In this system, a first step into the domain of multimodal languages and grammars. A heuristic, based on the minimum description length of the grammar, was developed in order to minimum description length of the multimodal grammar. However, the multimodal grammar description optimization through more learning operators are needed and different modalities can be adapted as per systems need.

References

[1] Arianna D'Ulizia, Fernando Ferri, Patrizia Grifon, “A Learning Algorithm for Multimodal Grammar Inference" *SYSTEMS, MAN, AND CYBERNETICS VOL. 41,NO. 6, DECEMBER 2011.*

[2] K. Nakamura and T. Ishiwata, "Synthesizing context free grammars from sample strings based on inductive CYK algorithm," *in Proc. 5th Int. Colloq. Grammatical Inference, 2000, pp. 186-195s*.

[3] K. Nakamura and M. Matsumoto, "Incremental learning of context free grammars," *in Proc. 6th Int. Colloq. Grammatical Inference,*
2002, pp. 174-184.

[4] K. Nakamura, "Incremental learning of context free grammars by extended inductive CYK algorithm," *in Proc. ECML Workshop Learn. Contex-Free Grammars, 2003, pp. 53-64.*

[5] K. Vanlehn and W. Ball, "A version space approach to learning contextfree grammars," *Mach. Learn., vol. 2, no. 1, pp. 39-74, 1987*.

[6] G. Petasis, G. Paliouras, V. Karkaletsis, C. Halatsis, and C. D. Spyropoulos, "*e-GRIDS:* Computationally efficient grammatical inference from positive examples,*" GRAMMARS, vol. 7, pp. 69-110, 2004.*

[7] J. Rissanen, "Modeling by shortest data description," *Automatica, vol. 14, no. 5, pp. 465- 471, Sep. 1978.*

[8] A. D'Ulizia, F. Ferri, and P. Grifoni, "Generating multimodal grammars for multimodal dialogue processing," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans, vol. 40, no. 6, pp. 1130-1145, Nov. 2010*

D. E. Knuth, "Semantics of context-free languages," *Theory Comput. Syst., vol. 2, no. 2, pp. 127-145, Jun. 1968.*R. J. Vidmar. (1992, August).