Multi way of input for effective communication.

Er. S.R.Gaidhani[1], Prof.Krishnakant Kishore[2]

[1]*M.tech Scholart Computer science, J.D.C.T.collage,Indore ,India*
[2] *Assistant Professor H.O.D.Department of Computer science, J.D.C.T.collage, Indore, India*

Abstract Human being can communicate with external entities with different modes like speech, text, facial expression, hand gesture. The use of this communicating ways makes human communication flexible with effectiveness. From last some decades years several techniques are used to bring close the human computer interaction as. It costs high for development and maintenance of Multimodal grammar in integrating and recognizing input in multimodal interfaces leads to investigate the novel algorithm which provides a way to automate grammar generation and grammar changing. In this we present a grammar algorithm that allows us to use a multimodal grammar from positive samples of multimodal approaches. The algorithm first create the multimodal grammar that is able to parse positive sample of sentences and by using learning operators and minimum length metrics for improving g multimodal grammar rammar description and avoids over generalization metrics.

***Index Terms*— Grammar Inference Algorithm, Over Generalization, Multimodal Grammar, Multimodal Sentence.**

I. INTRODUCTION

HUMAN COMMUNICATION is consist of multiple modes of signs, speech, touch, gesture etc. Interact in amongst people is carried through several communication channels i.e. multimodal communication. From last some decades several efforts have been made for communication of human computer interface by means of effectively with some concrete result.,. The purpose of this to make close & effective result in term of output. Therefore, multimodal interfaces allow us to communicate with computers through use of several channels or several mode of communication. Multimodal interaction provides the user with a way to interface with a system in both input and output, enabling users to communicate more easily with automated systems. The human-computer communication depends on the possibility of exchanging content through the communication methodologies or ways . Therefore, multimodal interfaces, which allow us to communicate with the computer through the simultaneous use of several channels of input/output at a single time, have gained increasing importance in human-computer interaction research. Multimodal grammars provide a widely used methodology [1]-[4] for integrating inputs by using multimodal interfaces.

In this technology, the outcomes of each unimodal recognizer are considered as terminal symbols of the grammar, and consequently, they are recognized by the parser as a unique multimodal sentence. Therefore, in the interpretation phase, the parser uses the specified grammar (production rules) to interpret each multimodal sentence in the input.

II. LITERATURE SURVEY

The studies in grammatical inference exist in several application domains. Such as speech recognition [10], computational linguistics [11], computational biology [12], [4], and machine learning[12][13]. Many of these learning models take as input an initial set of training examples and as output the language description, i.e., the specific grammar that accepts only those examples. Mostly algorithms for NL grammar inference focus on context free grammar. There are three existing grammatical algorithm discussed here.

The *inductive CYK algorithm [4]* synthesizes CFGs from positive and negative sample strings and generats the minimum production rules, which derive positive strings but do not derive any given negative strings. The main advantages of the extended inductive CYK algorithm rely on the generation of simpler sets of rules and shorter computational time (compared to the other grammatical inference mechanisms) in the inference of CFGs for some simple languages.

The *learning by version space algorithm* [14] needs positive and negative examples in inferring the grammar. The algorithm applies a particular induction technique, called as version space strategy, which is based on a compact way of representing the version grammars and some other processes have to choose among them grammars and some other processes have to choose among them.

The *e-GRIDS algorithm* [15] is a grammar inference method that uses positive training sentences in order to construct an initial grammar by converting each one of the training examples into a grammatical rule. Subsequently, the learning process, which is organized as a beam search, takes place. Having an initial hypothesis (the initial grammar) in the beam, 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, which facilitates its scalability to large example sets. Although this algorithm is able to infer grammars that perform well, based on relatively small sets of training examples, it is also able to handle large example sets in a significantly reduced amount of time.

The algorithm proposed in this paper combines the capabilities of inductive CYK and e-GRIDS algorithm

III. IMPLEMENTATION DETAILS

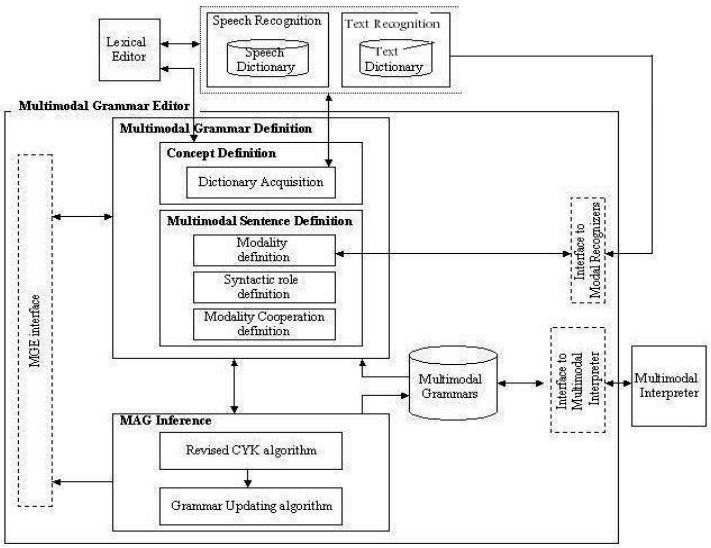

Fig 1: System Architecture

Using the Multimodal Grammar Editor, the language designer, which is the primary user of this component, can create all up the intended grammar or update an existing one. The MGE contains the Multimodal Grammar Definition and the MAG Inference components and a MGE interface. The MGE interface is a Multimodal User Interface (MUI) responsible of the interaction between the language designer and both the Multimodal Grammar Definition and the MAG Inference components. This interface allows the acquisition of the data to be used for inferring the grammar, i.e. the (positive) examples of sentences and the concepts used for expressing these sentences, since the grammar definition follows a by example approach. Moreover, it presents a view onto the multimodal grammar resulting from the MAG inference stage. The Multimodal Grammar Definition sets the grammar that the language de-signer wants to define by either instantiating a new grammar or selecting an existing grammar from the Multimodal Grammars repository, according to the designers choice. Furthermore, the Multimodal Grammar Definition is responsible for the linearization process, i.e. it takes the elements of the unimodal sentences, coming from the Multimodal Sentence Definition sub-component, and combines them opportunely, in order to generate a linear sequence of elements. Such a sequence represents the multimodal linearized.

A. Grammar Representation

In the proposed grammar inference 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 $X0 \rightarrow X1\ X2 \ldots Xn$ where

$n \geq 1$, $X0 \in N$ and $Xk \in N \cup T$ for $1 \leq k \leq n$

and $S \in N$ as a start symbol (or axiom);

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).

The attributes of the set $MS(X)$ are very general and independent from the application domain. They manage the multimodal properties of the sentence's symbols: value, modality, syntactic role, and modality cooperation type. This information is represented by the following four attributes of $MS(X)$.

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 (in our system, we have four 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*}*.

B. Multimodal Grammar Inference Algorithm

First Phase of the MGI 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, x_k , a set T = { x_1, $x_2 \ldots x_n$} of terminal symbols , a multimodal attributes grammar G = {G, A, R}. A target sentence x_t composed of terminal symbols $x_1 \in T$

Output:

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

Preconditions:

x is a string that has been parsed by the syntactic analyser yet, Each input element is then associated with a syntactic category n $\in N_0$.

Procedure:

(Generate a candidate set of production rules CPR used in step 2)

1. Consider x as the sentence x_1, $x_{2 \ldots\ldots}$ $x_{n(}$ the multiple combination)
   Generate the set P' of production rules that is composed of rules of the form A_i -> x_i
2. Continue the following processes for all $1 \leq i \leq k$
   i) Initialize a new CYK matrix C(k X k) by
   ii) Assign a value
   iii)Assign to each c_{ij} a set of functions.
3. 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 continue with step 2

Second Phase of the MGI Algorithm:

During the next phase, whose procedure, the analysis of the structures generated during the first phase is performed, which are the CYK matrix C and set CPR of candidate production rules. In particular, the algorithm selects the candidate derivations with the highest values. Nonterminal symbols, which belong to the set N_0, do not need any processing, while those symbols that are created during the first phase for simulating the generation of some productions need to be definitely included into the grammar. Consequently, non terminals that are part of the production rule inserted into the grammar have to be redefined until all symbols belong to the grammar. Therefore, the output of the MGI algorithm is a new MAG G'' = (G',A',R'),

Input:

An input sentence x : x_1, x_2, x_k , 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:

A new multimodal attribute grammar G' ={ G',A',R'} with G = (T',N',P',S') and R' = Rp U R'p

Preconditions:

The sentence x does not belong to the language by the current grammar G

Procedure:

(Generate a candidate set of production rules CPR used in step 2)

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 $\in$ 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.

Grammar Updating Steps of the MGI Algorithm:

The next step of the MGI algorithm is to update the MAG G', outputted by the first step, by evaluating its description length and by applying to it the learning operators to produce the equivalent grammar descriptions that are more "compact" with respect to the description length of the grammar.

Input:

A current multimodal attribute grammar G' = {G, A, R} with G = {T,N,P,S},A contains the sets of synthesized attributes S(x1) associated with each terminal symbol $x_i \in T$; R contains the semantic functions Rp for valuating the attribute of non-terminal in the head of some production rules in P'

Output:

A new multimodal attribute grammar is G''.

Procedure:

1. Evaluate the description length DL of G'
2. Iterate the following processes
   a. For each production p $\in$ P, such that p : A -> BC
   b. Evaluate DL of the new grammar G''
   c. For each production p $\in$ P, such that such that BC belongs to the body of p
   d. Evaluate DL of the new grammar G''

IV. EXPECTED RESULTS

Running of the First Step of the MGI Algorithm:

Consider the multimodal sentence composed of the speech "call that person" and text "Bob". The set T of terminal symbols is composed of the elements of each unimodal sentence,

T = {Call,that,person,Bob}.

The initial set of production rules P' contains the following rules:

P' = {VB -> Call; DT ->that; NN->person; NNS>Bob}.

Furthermore, the set of candidate production rules CPR contains the following rules:

CPR = {B -> VB DT; C -> DT NN;D -> NN NNS; E-> VB C; F ->B NN;G -> DT D;H -> C NNS; I -> VB G; L -> B D;M -> E NNS};

Improving the Grammar Description to Avoid the Over-Generalization Problem. The goal of the second step of the MGI algorithm is to update the MAG G', outputted by the first step, by evaluating its description length and by applying to it the learning operators to produce the equivalent grammar descriptions that are more "compact" with respect to the description length of the grammar.

The second step of the grammar inference method, named as the grammar-updating step, works in the following way.

It takes as input the MAG G' = {G',A',R'} generated during the first step, where G' = {T',N',P',S'}, A' = (A(X))X $\in$ N U T , and R' is the set of semantic functions for evaluating the attributes of X $\in$ N U T.

	Call	**That**	**Person**	**Bob**
	1	**2**	**3**	**4**
1	**VB** **VB.val <- Call** **VB.mod<-speech** **VB.synrole<-verb**	**DT** **DT.val <-That** **DT.mod <-speech** **DT.synrole<-deictic** **DT.coop <-compl.**	**NN** **NN.val <-person** **NN.mod<-speech** **NN.synrole<-noun** **NN.coop<-compl**	**NNS** **NNS.val <-atos** **NNS.mod <-text** **NNS.synrole<-noun** **NNS.coop <-compl**
2	**B** **B.val <-cal** **B.mod<-speech**	**C**	**D** **D.val <-Bob** **D.mod <-text**	
3	**E** **E.val<-call** **E.mod<-speech**	**G** **G.val <-Bob** **G.mod <-text**		
4	**I** **I.val <- call Bob** **I.mod <-SpeechText** **L** **L.val <-call Bob** **L.mos<-speech + text** **M** **M.val <- callBob** **M.mod<-speech_text**			

Table 1 :CYK Matrix

Multimodal Attribute Grammar

Here In proposed system following expected multimodal grammar has been shown by using speech and text modality.

P1) S -> NP VERB	P2) S->VP NP	P3) VP->VERBT
R1.1) S.val<-NP.val +VERB.val	R2.1)S.val<-VERBT.val	R3.1) VP.val<-VERBT.val
R1.2)S.mod<-NP.mod+VERB.mod	R2.2)S.mod<-VP.mod+NP.mod	R3.2) VP.mod<-VERBT.mod
P4) VERBT-> call	P5) NOUN -> Person	P6) NNP1 -> Bob
R4.1) VERBT.val<- call	R5.1) NOUN.val<- person	R6.1)NNP1.val<-Bob
R4.2) VERBT.mod <- speech	R5.2) NOUN.mod<- speech	R6.2)NNP1.mod<-text
R4.3) VERBT.synrole<- verb	R5.3) NOUN.synrole<- noun	R6.3)NNP1.synrole<-Bob
	R5.4) NOUN.coop<- complementary	R6.4)NNP1.coop<-complementary

V. CONCLUSION

Multimodal interaction has emerged in the last few years as the future paradigm of human-computer interaction. This fact is gathered also by the increasingly application of the multimodal paradigm to computer interfaces making computer behavior closer to human communication. Multimodal interaction requires that several simultaneous inputs, coming from various input modalities, are opportunely integrated and combined into a complete sentence. In order to overcome the deficiencies of the grammar based paradigm, this thesis proposed an approach of grammar definition that follows the "by example" paradigm, that is, the language designer provides concrete examples of multimodal sentences that have to be recognized, and a grammar inference algorithm automatically generates the grammar rules to parse those examples. In such a way no skilled grammar writers are needed, but even non-expert users can define multimodal grammars.

VII. REFERENCES

[1] Arianna D'Ulizia, Fernando Ferri, Patrizia Grifon. A Learning Algorithm for Multimodal Grammar Inference" *SYSTEMS, MAN, AND YBERNETICS 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. 186-195.*

[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.*

[9] D. E. Knuth, "Semantics of context-free languages," *Theory Comput. Syst., vol. 2, no. 2, pp. 127-145, Jun. 1968.*

[10] J. K. Baker, Trainable grammars for speech recognition," *in Proc. Speech Commun. Papers 97th Meeting Acoust. Soc. Amer., 1979, pp. 547–550.*

[11] P. Adriaans, "Language learning from a categorical perspective," *Ph.D. dissertation, Univ. van Amsterdam, Amsterdam, The Netherlands, 992.*

[12] Y. Sakakibara, "Recent advances of grammatical inference," *Theor. Comput. Sci., vol. 185, no. 1, pp. 15–45, Oct. 1997.*

[13] C. de la Higuera and J. Oncina, "Identification with probability one of stochastic deterministic linear languages," *in Proc. ALT, 2003, pp. 134– 148.*