The Incomplete Grammar approach to the development of a Strong-AI based ICALL system

Charles Grant Brown, Nathan Keim, Kevin Brammer, Lorne Flagel
Department of Computer Science, University of Northern British Columbia, Canada, V2N 4Z9
brownc@unbc.ca

Abstract

The development of our ICALL system is supported by the Incomplete Grammar approach, which allows us to model the learner with a cognitively based system of hypotheses held by the Tutor module. An Incomplete Grammar model of a learner's language performance includes Constituent Productions P_C, Constituent Production Errors P_{CE} , Lexical Productions P_{CL} and Lexical Production Errors P_{LE} in the learner model, but is incomplete in the target language space. The generation of the set of hypotheses of mal-rules (elements of P_{CE}, and P_{LE}) when language errors are detected is examined from a computational perspective. Dynamic evaluation and ranking of mal-rules is a challenge. We report on approaches to this problem. Strategies for further development are explored.

1. Introduction

In ICALL systems, the learner's knowledge of the domain is the learner's knowledge of the second language (L2). This knowledge is not directly accessible by an Intelligent Tutoring System but may be inferred from the learner's responses to learning and practice tasks. These responses will include grammatical and un-grammatical expressions. Both are useful in building and maintaining a model of the learner's knowledge of L2. The purpose of this modeling [1] is to base tutorial actions not only upon the immediate input, or response, of the learner to the learning task, but to extend the tutorial strategies to include judgments of the learner's knowledge of the whole domain.

It was shown in [2] that an incomplete grammar and lexicon can be inferred from the parsing of ungrammatical (and grammatical) input. Productions of the Incomplete Grammar are synthesized from the output of a robust parser. The synthesized productions become hypotheses in the learner model. A system of belief support may be used to maintain the learner model.

The long-term objective of this research has been to develop an ICALL system with a learner model firmly grounded in theories of language acquisition and to apply the model to the development of a language tutoring system.

2. Architecture of the system

The system conceptual architecture is a cognitive model. We build a "strong AI" model incorporating units that individually model the cognitive processes of the human tutor in making hypotheses about the learner's knowledge of L2. The Tutor module consists of three subsystems: Hypothesis Generation, Tutoring Strategy and Dialogue Management.

The grammar G_{L2} is based on the unification grammar approach. The parser is a variation of standard chart parsers, but one which interfaces seamlessly with the Hypothesis Engine module of the Tutor. Questions of computational complexity are being addressed in order to make the process time- and space-efficient.

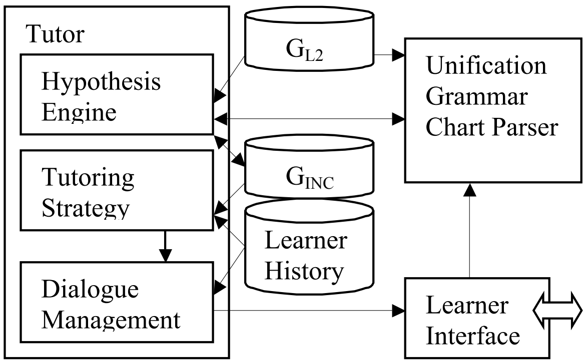

Diagram 1: Architecture of the System

The question of choice of tutorial strategy is largely un-addressed. An ad hoc approach with pre-assigned tutorial strategy has been used as the architecture is tested.

3. Knowledge of Language

3.1 The language competence of L2 learners.

An Incomplete Grammar (G_{INC}) is a dynamic model of the learner's knowledge of L2. As new productions (grammar rules) and lexical items are used by the learner, the Incomplete Grammar is updated dynamically. This model supports a view of incremental acquisition of language, including the acquisition of correct and incorrect productions. As such, a G_{INC} is a dynamic model of the learner's competence. The grammar of focus in ICALL is the grammar of the language L2 which is denoted by G_{L2}. This is assumed to be a complete grammar, meaning that G_{L2} is correct and covers all constructions of L2 of interest to the tutoring system.

However, the learner's knowledge must be assumed to be incomplete. The learner's knowledge of L2 is modeled as an Incomplete Grammar. An Incomplete Grammar G_{INC} is defined to be the composition of the four disjoint sets: Constituent Productions (P_C), Constituent Production Errors (P_{CE}), Lexical Productions (P_{CL}), and Lexical Production Errors (P_{LE}), where the constituent productions, P_C are the subset of interest of the productions of G_{L2}, and the lexical production, P_{LE}, are the subset of interest of the lexical productions (or lexicon) of G_{L2}.

The core idea in our system is the hypothesis of learner errors. Constituent production errors, P_{CE} are the hypothesized production errors of the learner, and the lexical production errors P_{LE} are the hypothesized lexical errors of the learner. These two sets, P_{CE} and P_{LE}, are disjoint respectively from the constituent productions P_C and the lexical productions P_{LE}.

It is our observation that second language learners understand the concepts of constituents such as noun phrase, prepositional phrase and verb form (inflected and compound). The difficulty for a language learner is in two tasks; constructing the constituent itself and syntactically locating the constituent.

We model the cognitive process of the human tutor as a process of synthesizing hypotheses of errors the language learner makes in performing these two tasks.

Minimalist theory [3] recognizes that an explanatory model of language acquisition must account for learning language in a short period of time and must account for errors in the learning process of acquiring a rich lexicon and rules of syntax. This is accounted for through a process of acquisition of principles (e.g. barriers) and parameters (e.g. concordance constraints). The Incomplete Grammar model recognizes this view, and attempts to explain language acquisition as a process of hypothesis formation, successful and unsuccessful, based on unification of feature values and productions.

3.2 The synthesis of G_{INC} productions

An example of hypothesis formation for G_{INC}, extra word error, is presented here. Additional examples and a more extensive discussion can be found in [2]. This example illustrates the general algorithmic methodology for the synthesis of the hypotheses in G_{INC}. It also illustrates the interface between Chart Parser and Hypothesis Engine modules:

Extra word error: In this case, an extra word appears in the parser results, preventing further parsing. The extra word may appear before or after a word it is intended to modify. There are several sub-cases of this phenomenon, each involving a different adjacent known constituent. One of these is discussed here, a case where the extra word is an unnecessary specifier (article, in this case) of the XP involved. The algorithm for hypothesis synthesis is:

1. Where a constituent has been parsed which cannot be combined with surrounding constituents to produce a larger constituent, and where the following pattern pertains:
   - Contiguous constituents $\beta_j = P_j F_j$ and $\beta_k = P_k F_k$ have been produced by the chart parser.
   - A production does exist in G_{L2} of the form:
     $\alpha \rightarrow \beta_{k2}$

   where:
   $\alpha = P_\alpha F_\alpha$ has P_α = XP where XP is bound to some category in {np,pp,vp,ap},
   and the feature list F_α contains [cat:X]
   and where:
   $\beta_{k2} = P_{k2} F_{k2}$ has P_{k2} = Xbar where Xbar is bound to the corresponding category in
   {nbar, pbar, vbar, abar}
   and the feature list F_{k2} contains the pair [cat:X]
   and where: β_k does unify with β_{k2}
   Infer the production $\alpha \rightarrow \beta_j \beta_{K2}$ as constituent production error in the set P_{CE}. (This production becomes a hypothesis of the learner's knowledge of L2 which may be added to the set P_{CE} with a confidence factor yet to be determined.)
2. Recursively apply the algorithms to infer all possible new productions.
3. Resume chart parsing with the augmented incomplete grammar.

As an example, consider a situation where the learner has produced the phrase *[I am traveling to the Washington]*. This contains an extra article *[the]*. The relevant constituent productions P_C of grammar L2 near the extra word are:

```
pp( […])  --> prep([…]),  np([…]).
np([nbr:N, type:proper, def:yes])  -->
    nbar(nbr:N, type:proper, def:yes].
nbar([nbr:N,type:proper]) -->
    noun([nbr:N,type:proper]).
```

The grammar L2 lexical productions P_{CL} are:

```
prep( […]) --> [to].
det([…]) --> [the].
```

noun([type:proper, nbr:sing]) --> ['Washington'].

From the phrase *[to the Washington]* produced by the learner, the parser will produce the three constituents of a partial parse:

prep(to):[…]
det(the):[nbr:sing,gndr:masc]
noun(Washington):[nbr:sing,def:yes, type:proper]
nbar(noun(Washington)):[nbr:sing, def:yes, type:proper]

we can infer the production:

np([nbr:N, type:proper, def:yes]) -->
det(nbr:N, def:yes),
nbar(nbr:N, type:proper, def:yes].

from the analysis of the partial parse, and add it to the incomplete grammar production errors P_{CE}. Thus, G_{INC} is dynamically changed as a result of the input. The augmented G_{INC} may then be used to guide the tutorial strategy._The algorithm given above for the inference of mal-rules is one of a set of algorithms. Collectively, they provide a set of extended rules of inference. They are sufficient to synthesize hypothesis about a broad range of learner errors. The learner error hypotheses realized are themselves developed from specific errors. Some generality is achieved through the feature lists in the synthesized hypotheses.

3.3 Grammars for G_{INC} implementation

Definite Clause Grammar (DCG) notation is used. Unification grammar notations of various types (LFG, HPSG etc.) are readily encoded as DCGs [4]. Productions (e.g. NP → Det Nbar) have long been augmented with features in Linguistics literature. In DCGs we add a list of feature:value pairs to the production predicates (constituent names) and to the lexical entries. DCGs support the idea of a syntax and a grammar driven by a rich lexicon.

np([category:np, case:accusative, number:plural, gender:feminine, person:third, definite:yes, type:pronoun])	det(a, [category:det, definite:no, number:singular, gender:G, person:third]) --> [a].
(a)	(b)

Table 1: DCG encoding of (a) production predicate and (b) lexicon entry

The use of a richly featured lexicon and productions makes it possible, in turn, to hypothesize production errors using the algorithms of section 3.

3.4 Relative importance of G_{INC} productions

When a human tutor makes a hypothesis about the learner's grammatical error, a level of confidence is associated with that hypothesis. The tutor can use the level of confidence in an error as a factor in deciding what tutorial strategy (e.g. focus on form, Socratic question, positive/negative feedback, reply with correction) to employ. As evidence accumulates, the tutor becomes more (or less) certain about individual hypotheses.

4. The Tutorial System

A prototype tutorial system has been built to take advantage of incomplete grammars. The system was tested on control and experimental groups [5] [6] to validate the model and to measure learning directly.

4.1 Grammatical coverage

The tested coverage of the tutoring program consists of lessons on twelve English verb tenses: simple present, simple past, simple future, present progressive, past progressive, future progressive, present perfect, past perfect, future perfect, present perfect progressive, past perfect progressive, and future perfect progressive.

4.2 Learning exercises

For each of the activities, the learner is given a lesson on the verb tense that is going to be tested. The learner is then presented with the learning exercises.

4.3 Tutoring Strategies

The tutoring strategies may not always give a recommendation on which question to ask next. Questions have a pre-set (default) ordering so that there is always a question available. This default structure is the only tutoring strategy used for control groups. However, the program also gives a simple correct/incorrect response to the subjects' responses. The other tutoring strategies are based on the learner model. They use the information contained to influence the choice of the next question. The process of selecting the next question to present to the learner is only partially based on recommendations provided by the tutoring strategies. More than one follow-up question may be valid. The *Question Selector* module, chooses which to present. Also based on recommendations provided by the tutoring strategies, the Question Selector may find it appropriate to move on to the next lesson.

4.4 Tutorial Intervention

Like a human tutor, the system does accept different versions of the correct answers due to the grammatical coverage provided. The ability to tell the subject whether he/she is correct or not, is actually more sophisticated than a textbook, even for the control group.

For the main experimental group, if the answer is incorrect, the learner is told whether the tense used is correct or not and whether the learner has made a similar mistake before. The system says if the answer may have a grammatical problem, says if there is a word missing in the verb phrase, and says whether the learner should use hints or start the lesson over.

5. The Learner Model

The Learner Model contains the Learner History and G_{INC}. All responses are saved, as well as the corresponding hypothesized mal-rules that are used on the subjects' answers. The Learner Model contains which questions are answered correctly/incorrectly/skipped, the order in which they were answered, and the mal-rules used. This produces a confidence score used to help determine the next course of action for the program. Starting and finishing times are also recorded.

5.1 Productions

The Learner Model includes hypothesized mal-rules (production errors including mal-lexical productions) used and which words entered by the learner were unknown to the system. Which rules are used, how many times they are used, and what questions they are used for all appear in the model.

6. A Study of Learner Performance

Statistical tests [5] for significance (t-tests) examined (a) the difference between scores on the pre-test and post-test and (b) the time each of the two groups spent compared to how far they proceeded. The program did not appear to significantly increase the English ability of the experimental group. The program did significantly increase the time/distance ratio of the experimental group.

7. Discussion

Our results show us that it is possible to build a cognitively modelled ICALL system. We have attempted to combine cognitive models of language acquisition with our knowledge of tutorial strategies and learner modelling. The system incorporates AI methods of machine learning and dynamic modelling of the user.

Many questions remain to be examined:

- Correlation of tutorial strategies based on sound and accepted pedagogical models must be explored and incorporated. This is a very complex issue, yet one which must be undertaken to approach the competence of human tutors.
- Representation of grammars must be examined from the point of view of the learner. The issue here is the correlation of the sophisticated unification grammar productions with the learner's view of language and grammar.
- Time and space complexity of the Hypothesis Engine must be further examined. Heuristic approaches to hypothesis formation may be useful.
- The development of metrics for the evaluation of ICALL systems. In particular, more direct controlled experimentation is required. Grammatical coverage of tutoring can be expanded and associated issues examined.
- Continued experiments with to evaluate results of prototype systems against control groups.
- The possibility of tailoring our approach to childhood learners of L2.

The Learner Model is necessarily complex. This is not a disadvantage, but as a natural outcome of the complexity of language and language learning. The use of this type of dynamic learner model is appropriate to models of language acquisition.

We have discussed in [2] several complexity-theoretical questions in the synthesis of hypotheses. We continue to study these questions.

References:

[1] Etienne Wenger, *Artificial Intelligence and Tutoring Systems*, Morgan-Kaufmann Publishers, 1987.

[2] Charles Grant Brown, *Inferring and Maintaining the Learner Model,* Computer Assisted Language Learning, Volume 15, No. 4, 2002.

[3] Noam Chomsky, *The Minimalist Program*, MIT Press, 1995.

[4] W.F.Clocksin & C.S. Mellish. (1984). Programming in Prolog, 2nd ed., Spring Verlag, 1984

[5] Nathan Keim. (2003).*A DiverseUserModel in the Context of an Intelligent Tutoring System*, M.Sc. Thesis, University of Northern British Columbia.

[6] Charles Grant Brown and Nathan Keim, *Assessing Learner Models generated with an Incomplete-Grammar Based CALL system*, in Proceedings of PEG 2003, St. Petersburg, Russia.