

Automatic Analysis of Learner Language for ICALL

L485/L700

Dept. of Linguistics, Indiana University
Autumn 2008

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

1 / 26

Automatic Analysis of Learner Language

Where we're going today:

- ▶ ICALL context
- ▶ Learner corpora & learner properties
- ▶ Developing taxonomies
- ▶ Parsing & automatic analysis
- ▶ From parsing to features
- ▶ Some practical points

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

2 / 26

Guiding Questions for the Semester

- ▶ Which properties of learner language are useful and relevant to obtain for Foreign Language Teaching (FLT) and current Second Language Acquisition (SLA) research?
- ▶ What annotation scheme or (error) taxonomy is appropriate for this and how do different annotation schemes compare?
- ▶ How reliably can errors and other properties of learner language be obtained automatically given the current state-of-the-art in NLP?
- ▶ What is the impact of the specific properties of learner language on the (re)use of NLP technology? How does it impact performance and the potential use of such technology in foreign language teaching tools?

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

3 / 26

The context

One of the first questions to ask is: for what purpose are we analyzing learner language?

The focus of this course—but by no means the only focus—is:

- ▶ How can we automatically analyze learner language in order to provide feedback to a learner?
- ▶ Although this can be done under many settings, we'll look at how it fits into ICALL systems

This requires at least two perspectives:

- ▶ How does one describe a linguistic construction in a descriptively-adequate way, such that it supports feedback for learners? (first half of the semester)
- ▶ What are automatic techniques for finding & diagnosing non-targetlike uses of constructions? (second half)

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

4 / 26

What is ICALL?

Definition of CALL (Heift and Schulze (2007, p. 7), from Levy (1997)):

- ▶ "Computer-Assisted Language Learning (CALL) may be defined as 'the search for and the study of applications of the computer in language teaching and learning'"

Intelligent CALL (ICALL), or parser-based CALL (Heift and Schulze (2007, p. 4), from Holland et al. (1993)):

- ▶ ICALL relies on parsing, "a technique that enables the computer to encode complex grammatical knowledge such as humans use to assemble sentences, recognize errors and make corrections"

As we will see later, we may or may not need an actual parser

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

5 / 26

Form-focused instruction

Some believe all that is needed for learning is "comprehensible input" (Krashen 1982) (cf. Nerbonne 2003)

- ▶ If comprehensible input is all that is needed, form-focused instruction as in ICALL is unnecessary

However, that is not the whole story:

- ▶ DeKeyser (1995, 2000): adult learners benefit from explicit instruction
- ▶ Levy (1997): CALL should be motivated by practical considerations, not theory

Even in the context of ICALL, we will want to focus on general techniques for analyzing learner language

- ▶ Analyzing learner language does not have to be for (a particular version of) form-focused instruction

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora

Useful text types

Useful types of annotation

Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

6 / 26

Benefits of CALL

Holland et al. (1993) outline some benefits of CALL:

- ▶ As a supplement to in-class learning, (I)CALL can provide form-focused instruction and foster language awareness
 - ▶ Students allowed to make their own errors and have them pointed out
- ▶ If foreign language teaching is unavailable, inconvenient, or unaffordable, (I)CALL can be used.
- ▶ (I)CALL can be used for research
 - ▶ Automatically track student responses & types of errors made, which can provide information about patterns of acquisition
 - ▶ Track preferences for modalities, information sources (e.g., dictionary), teaching methods, etc.
 - ▶ Provide more data to investigate questions about learner language & development

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

The need for NLP

- ▶ Existing CALL systems which offer exercises
 - ▶ typically are limited to uncontextualized multiple choice, point-and-click, or simple form filling, and
 - ▶ feedback usually is limited to yes/no or letter-by-letter matching of the string with a pre-stored answer.
 - ▶ An example for letter-by-letter feedback on the "Spanish Grammar Exercises" site (Barbara Kuczun Nelson).

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

From CALL to ICALL

- ▶ Linguistic modeling is needed to improve on this situation, e.g.:
 - ▶ tokenization: identify words
 - ▶ morphological analysis: identify/interpret morphemes
 - ▶ syntactic analysis: identify selection, government and agreement relations and word order requirements
 - ▶ formal pragmatic analysis: identify coreference relations, information structure partitioning, ...
- ▶ Computational tools identifying such linguistic properties need to be integrated into CALL systems to obtain language-aware "Intelligent" CALL (ICALL).
 - ▶ Tools must be extended/written to permit and diagnose errors made by language learners.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

Limitations of NLP

Unlike classroom situations, where the instructor can be assumed to be right, ICALL systems are prone to errors

- ▶ If parsers or other NLP tools return wrong analyses, learners could be told they are right when they are wrong, or vice versa
- ▶ The emphasis is generally on *precision*: make sure the cases we say are errors are truly errors
 - ▶ When analyzing learner language to provide feedback, we do not want the feedback to be incorrect; no feedback is better

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

Learner corpora

Learner corpora are texts of learner language, e.g., Chinese learners of English

- ▶ They may be unannotated, or they may be annotated with properties about learner language
- ▶ Annotation can help determine which language properties are underused, overused, or misused.

Creating a learner corpus is a major undertaking and gets into issues like:

- ▶ representativeness & appropriateness to a task

For automatic analysis of learner language, a crucial question is: what kind of annotation is needed?

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

Annotated learner corpora: Starting point

- ▶ Annotated learner corpora can
 - ▶ help validate generalizations about language acquisition
 - ▶ provide a broad empirical basis for the development of new hypotheses and theories
 - ▶ inform foreign language teaching practice
- ▶ Depending on the corpus, they can support
 - ▶ qualitative and quantitative analysis
 - ▶ including longitudinal analysis
- ▶ To play these roles, the terminology used to single out the learner language aspects of interest needs to be mapped to instances in the corpus.
- ▶ Effective querying of corpora typically requires reference to annotated abstractions (linguistic classes, errors) instead of extensionally characterizing individual strings.

Thanks to Detmar Meurers for this & the next 5 slides

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
 Useful text types
 Useful types of annotation
 Ambiguity and representation
 Developing error taxonomies
 Parsing & automatic analysis
 Feature-based analysis
 Some practical points

<h3>Issues with learner language corpora</h3> <p>The types of learner texts included in corpora</p>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<ul style="list-style-type: none"> ▶ Most learner language corpora consist of essays. ▶ Yet in a typical communicative or task-based language learning setup (such as an ICALL system), learners produce language in a wide range of activities, e.g., <ul style="list-style-type: none"> ▶ answering reading or listening comprehension questions ▶ asking questions in information gap activities <p>⇒ To obtain corpora representative of learner language, it would be advantageous to include language produced in a variety of language activities.</p> <ul style="list-style-type: none"> ▶ Including explicit task contexts in the meta-information of a corpus can also provide constraining information useful for interpreting learner language. <ul style="list-style-type: none"> ▶ e.g., it's easier to infer what a learner wanted to say if one knows the text they are answering questions about. 	<p>13 / 26</p>

<h3>Issues with learner language corpora</h3> <p>The types of annotations provided</p>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<ul style="list-style-type: none"> ▶ The annotation of learner language has typically focused on errors made by the learners. ▶ At the same time, learner errors often are correlated with <ul style="list-style-type: none"> ▶ specific linguistic environments (classes, constructions)? ▶ specific language tasks performed by the learner (e.g., answering reading comprehension questions)? ▶ or specific strategies needed to complete particular tasks (e.g., skimming, scanning)? 	<p>14 / 26</p>

<h3>Issues with learner language corpora</h3> <p>The types of annotations provided (cont.)</p>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<ul style="list-style-type: none"> ▶ Linguistic aspects of learner language are relevant for SLA research and FLT independent of errors: <ul style="list-style-type: none"> ▶ overuse or underuse of specific language patterns. ▶ measures of language development (production, understanding), e.g.: <ul style="list-style-type: none"> ▶ Developmental Sentence Scoring (Lee 1974) ▶ Index of Productive Syntax (Scarborough 1990) ▶ Developmental Level (Rosenberg and Abbeduto 1987; Covington et al. 2006; Lu 2008) ▶ Finding the relevant patterns and computing the measures requires general linguistic annotation. <ul style="list-style-type: none"> ⇒ Learner corpora should ideally provide such annotation in addition to the error annotation. 	<p>15 / 26</p>

<h3>Issues with learner language corpora</h3> <p>Ambiguity and representation</p>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<ul style="list-style-type: none"> ▶ An error annotation scheme needs to support <ul style="list-style-type: none"> ▶ unambiguous and consistent identification of error <ul style="list-style-type: none"> ▶ generally involves identification of target intended by learner ▶ a unique representation of the identified error ▶ Annotation scheme design thus requires answering questions such as: <ul style="list-style-type: none"> ▶ Where can which ambiguities be reliably resolved, given what ling. context or other information (learner, task)? ▶ In a hierarchical tagset (i.e., different levels of specificity) how is consistency of level of annotation achieved? <p>⇒ Only distinctions reliably identified given information present in a corpus or its meta-information should be included in an annotation scheme.</p>	<p>16 / 26</p>

<h3>Issues with learner language corpora</h3> <p>Ambiguity and representation (cont.)</p>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<ul style="list-style-type: none"> ▶ Identifying the nature of the error <ul style="list-style-type: none"> ▶ Example: <i>The man eat cheese.</i> <ul style="list-style-type: none"> ▶ agreement error: <i>The man_{3s} eat_{not(3s)}} cheese.</i> ▶ tense error, intended was: <i>The man ate cheese.</i> ▶ Localizing and representing the error <ul style="list-style-type: none"> ▶ Which single, unique way is chosen to <i>annotate</i> an identified error, e.g., for binary relations? ▶ Example for marking a subject-verb agreement error: <ul style="list-style-type: none"> ▶ on the subject: <i>The man eat cheese.</i> ▶ on the verb: <i>The man eat cheese.</i> ▶ on an annotated relation: <i>The man →_{agr} eat cheese.</i> ▶ Problem is non-trivial given that <ul style="list-style-type: none"> ▶ suffixes in fusing languages combine multiple features (e.g., person, number, gender, case) ▶ often multiple relations are established (e.g., D-A-A-N) 	<p>17 / 26</p>

<h3>Developing error taxonomies</h3>	<p>Auto. Analysis of Learner Lg.</p> <p>ICALL context</p> <p>Learner corpora</p> <p>Useful text types</p> <p>Useful types of annotation</p> <p>Ambiguity and representation</p> <p>Developing error taxonomies</p> <p>Parsing & automatic analysis</p> <p>Feature-based analysis</p> <p>Some practical points</p>
<p>As alluded to, errors may be grouped hierarchically and may be categorized in different ways in an error taxonomy.</p> <ul style="list-style-type: none"> ▶ Again, a guiding question is: how reliably and thoroughly can a taxonomy be employed? <p>Error taxonomies can differ in:</p> <ul style="list-style-type: none"> ▶ How much linguistic information is encoded in an error type <ul style="list-style-type: none"> ▶ Possible error types for <i>Man ate cheese</i>. (where it is clear the correction is <i>The man ate cheese</i>.: <ul style="list-style-type: none"> ▶ Surface level: Omission error ▶ Lexical level: <i>the</i> error ▶ Part-of-speech (POS) level: Determiner error ▶ Syntactic level: Noun phrase formation error ▶ Semantic/Pragmatic level: Reference (?) error 	<p>18 / 26</p>

Developing error taxonomies (cont.)

Error taxonomies can differ in:

- ▶ To what degree errors are defined in terms of their corrections
 - ▶ Why corrections are difficult:
 - ▶ They sometimes miss the point: "Omitted word" may not be a useful designation.
 - ▶ They are not always clear: *Man ate cheese.* could also be corrected to *Men ate cheese.*
 - ▶ Why corrections are useful nonetheless:
 - ▶ If a preposition is omitted, this seems to be quite a different error than if a different one is used.
 - ▶ Both for identifying consistent learner problems and for evaluating systems, it can help to know, e.g., which prepositions a learner is consistently not using.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

19 / 26

Developing error taxonomies (cont.)

Error taxonomies can differ in:

- ▶ How cross-linguistically relevant they are
 - ▶ Are word order errors the same type of error in a language with some degree of free word order? Or a language like German, with V2 constructions?
 - ▶ Are Korean postposition errors the same as English preposition errors? What elements of an error taxonomy can carry over?
- ▶ How specific they are to a given context:
 - ▶ If a system will only provide feedback on morphology, then other layers of linguistic annotation seem to be irrelevant

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

20 / 26

Parsing & automatic analysis

The "intelligent" part of ICALL has sometimes been defined as having a parser to provide linguistic abstraction to the learner's language

- ▶ For many ICALL systems, parsers should:
 - ▶ Assign an analysis to a learner's input;
 - ▶ Identify erroneous spots in the input;
 - ▶ Provide an analysis of what the error signifies.
- ▶ More generally, NLP tools can provide linguistic "annotation" on learner data, identifying relevant language properties for feedback.

The crucial part of the "intelligence" seems to be in providing appropriate and accurate feedback to a learner.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

21 / 26

Detecting language misuse

A traditional way to look at ICALL is:

- ▶ You have a parser to assign grammatical analyses to sentences
- ▶ Learner language is *ill-formed*, so the parsing architecture has to be modified
 - ▶ Include **mal-rules**, rules to handle error cases.
 - ▶ e.g., A singular noun and a plural verb are allowed to combine, marked as an error.
 - ▶ Perform **constraint relaxation**, i.e., rework a parser to accept ill-formed input.
 - ▶ Some constraints (e.g., a subject and verb must match in number) are relaxed while parsing

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

22 / 26

Main problems with parsers for CALL

- ▶ They have the potential to return incorrect analyses, which is true of virtually any NLP technology
- ▶ The parser's analysis may be irrelevant or difficult to determine
 - ▶ If a parser returns only constituencies, without grammatical functions, it is much more difficult to distinguish prepositional uses.
 - ▶ Some ambiguities may be impossible to determine or may not really matter, e.g.,
(1) I saw the mouse [in the house] [by the garden].
- ▶ Syntactic analysis may not be needed, e.g., for morphologically-rich languages or for identifying semantic priorities
 - ▶ A student who is trying to communicate meaning may need feedback on that meaning.
 - ▶ Some form of (domain-specific) semantic analysis may be necessary

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

23 / 26

From parsers to features

In many situations and for many languages, we may:

- ▶ Not have the luxury of a parser (or one which gives us desired linguistic properties in an efficient manner)
- ▶ Need to maintain general methods which extend to a variety of ICALL domains
- ▶ Not fully understand what makes the selection of a correct language construction correct

For these reasons, we will explore some machine learning approaches to analyzing learner language this semester

- ▶ These require us to specify appropriate features that the machine can learn from.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

24 / 26

Potential pitfalls of machine learning

- ▶ Feature selection is non-trivial and will dramatically affect the outcome of your system
 - ▶ Often we have to approximate syntactic parses in the features, e.g., following noun \approx object NP
 - ▶ Features can actually be extracted from parses
- ▶ We have to understand enough about machine learning to use appropriate settings, without becoming machine learning experts
- ▶ We still need to be able to provide feedback which relates to grammatical terms learners know
 - ▶ If we detect an error, does the machine learner output give us enough information to say anything useful?
 - ▶ Would we still need a parse to provide feedback?

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

25 / 26

Some practical points

Based on the preceding discussion, we need to do the following this semester:

- ▶ Familiarize ourselves with the types of distinctions that are encoded in the annotation of learner language and for what purposes they serve
 - ▶ How does one obtain or create learner corpora?
 - ▶ What are good annotation schemes to use or create for unannotated data?
 - ▶ Practically speaking, how does one work with such corpora, i.e., what tools are available to ease the annotation process?
- ▶ Determine what properties of language predict correct/incorrect usage of language constructions
 - ▶ What NLP tools can be re-used for different languages, and how can they be re-used?
 - ▶ What features need to be extracted, and how can they be extracted?
 - ▶ Practically speaking, how do we get our data in the right format, run the NLP tools, and run a machine learning system?

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

26 / 26

References

- Covington, M. A., C. He, C. Brown, L. Naci and J. Brown (2006). *How complex is that sentence? A proposed revision of the Rosenberg and Abbeduto D-Level Scale*. CASPR Research Report 2006-01, The University of Georgia, Artificial Intelligence Center, Athens, GA.
- DeKeyser, R. (1995). Learning Second Language Grammar Rules: An Experiment with a Miniature Linguistic System. *Studies in Second Language Acquisition* 17(3), 379–410.
- DeKeyser, R. (2000). The Robustness of Critical Period Effects in Second Language Acquisition. *Studies in Second Language Acquisition* 22(4), 499–533.
- Heift, Trude and Mathias Schulze (2007). *Errors and Intelligence in Computer-Assisted Language Learning: Parsers and Pedagogues*. Routledge.
- Holland, V. M., R. Maisano, C. Alderks and J. Martin (1993). Parsers in Tutors: What Are They Good For? *CALICO Journal* 11(1), 28–46. <http://calico.org/journalarticles/Volume11/vol11-1/Holland,etal.pdf>.
- Lee, L. (1974). *Developmental Sentence Analysis*. Evanston, IL: Northwestern University Press.
- Levy, Michael (1997). *Computer-Assisted Language Learning: Context and Conceptualization*. New York: Oxford University Press.
- Lu, Xiaofei (2008). Automatic measurement of syntactic complexity using the revised developmental scale. In *Proceedings of the 21st International Florida Artificial Intelligence Research Society Conference (FLAIRS-08)*. Coconut Grove, FL: AAAI Press.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

26 / 26

Nerbonne, John (2003). Natural language processing in computer-assisted language learning. In Ruslan Mitkov (ed.), *The Oxford Handbook of Computational Linguistics*, Oxford University Press. <http://www.let.rug.nl/~nerbonne/papers/nlp-hndbk-call.pdf>.

Rosenberg, S. and L. Abbeduto (1987). Indicators of linguistic competence in the peer group conversational behavior of mildly retarded adults. *Applied Psycholinguistics* 8, 19–32.

Scarborough, Hollis S. (1990). Index of Productive Syntax. *Applied Psycholinguistics* 11(1), 1–22.

Auto. Analysis of Learner Lg.

ICALL context

Learner corpora
Useful text types
Useful types of annotation
Ambiguity and representation

Developing error taxonomies

Parsing & automatic analysis

Feature-based analysis

Some practical points

26 / 26