The Computer and Natural Language (Ling 445/515)

Computer-Aided Language Learning

Markus Dickinson
Dept. of Linguistics, Indiana
Autumn 2010
Some common computer uses

- Computers are widely used in support of foreign language teaching (FLT). For example, they
  - provide access to foreign language newspapers, radio, and TV programs through the world-wide web
  - connect language learners with native speakers through email/chat
  - support multimedia presentations providing an audio-visual foreign language context
  - enable the learner to search for real-life examples in electronic corpora
- Essentially, such computer usage helps language learners experience a foreign language and culture in a more direct, real-life fashion.
Awareness of language forms is important

- Which other aspects of language learning are relevant in this context, apart from the undisputed importance of contextualized, communicative language usage?

- Research since the 90s has shown that awareness of language forms and rules is important for an adult learner to successfully acquire a foreign language.
  - CALL can provide an opportunity to enhance awareness of a language’s rules
Real-life constraints

- The time a student can spend with an instructor/tutor typically is very limited.

- In consequence, work on form and grammar is often deemphasized and confined to homework so that the time with the instructor can be used for purely communicative activities.

- The downside is that the learner has relatively few opportunities to gain awareness of forms and rules and receive individual feedback on errors.
An opportunity for CALL

- The situation seems like an excellent opportunity for developing Computer-Aided Language Learning (CALL) tools to
  - provide individual feedback on learner errors and
  - foster learner awareness of relevant language forms and categories.

- But 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.
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.

CALL is a big business: 106 million Euro (about $120 million) spent on CALL products in Europe in 1994, and US market is twice as big. (Nerbonne 2003)
Expectations

How successful we think CALL systems are depends upon our expectations and upon the expectations of the students.

- Unlike classroom situations, where the instructor can be assumed to be right, ICALL systems are prone to errors:
  - Parsers return wrong analyses: students could be told they are right when they are wrong, and vice versa
  - Parsers are incomplete: no analysis may be returned

- Students have to understand up front that this is not a human:
  - Perhaps obvious, but affects, for example, the image of a tutor you might have on screen
Second Language Acquisition (SLA)

If we want to design a system to teach, e.g., English progressives to Japanese speakers, we have to ask:

▶ What kinds of exercises are best for learning this construction, or for any sort of grammatical principle?
▶ How advanced are these learners? Should we even bother teaching this to novice learners?
▶ Do Japanese speakers have trouble with this construction, or is it pretty easy to grasp?
▶ Is a given individual generally a good student? Do they have problems with certain constructions?

Insights from research on second language acquisition (SLA) can provide a starting point to answer these questions.
CALL Systems

Having discussed what we need to know when learning a language, we can now focus on:

- Basic uses of computers for CALL
- Early CALL Systems
- Intelligent CALL (ICALL)
  - Straightforward uses of NLP for CALL
  - Authentic Text ICALL Systems
  - Parser-Based ICALL Systems
Basic uses of computers for CALL

Lots of general possibilities for using a computer to learn:

▶ multimedia presentations
▶ online dictionaries with fast access
▶ extensive databases of information
▶ digital audio files
▶ digital videos of people speaking in L2
  ▶ Digital advantages: easy playback, easy isolation of problematic spots, etc.
▶ interactive games & puzzles
▶ exercises for students to complete

The last two examples potentially require sophisticated natural language processing
Early CALL systems

Some older types of systems are along the following lines

▶ **Frame-based systems** “match student answers with a set of correct and incorrect answers stored in a frame”

These systems differ in their strategies for selecting questions, but they rely on preset questions and answers

▶ In principle, could be used with NLP techniques
A **linear system** does the following:

1. poses a question
2. accepts an answer
3. informs the student as to whether or not the answer was correct

⇒ Regardless of the correctness of the answer, linear systems proceed to the next question.

- **Branching systems** select a question of appropriate difficulty based on performance thus far
Problems with frame-based systems

Frame-based systems are fairly simple and generally do not involve much linguistic knowledge

▶ There is no deep understanding of question domain
▶ They generally only match answers with questions, but language use is more varied
  ▶ Could be more than one correct answer, as with a translation task
  ▶ Fill-in-the-blank exercises (pointed out by Trude Heift):

  (1) Today is November 10. What date is tomorrow? Tomorrow is _____.
    a. The eleventh. [Judged correct]
    b. November 11. [Judged incorrect]

▶ There is not much tailoring to particular student needs
Intelligent CALL (ICALL) focuses on using linguistics and natural language processing to make CALL better.

- Some basic NLP uses for CALL (Nerbonne 2003):
  - Concordancers
  - Text alignment
  - Speech recognition and synthesis
  - Morphological processing (for vocabulary help)

- ICALL can also involve integrating authentic text into exercises, usually for more advanced learners

- ICALL involves using a parser (syntactic processor) to handle real learner input and deals with issues like:
  - Diagnosing and accounting for user errors
  - Modeling the system on particular (kinds of) users
  - Presenting useful feedback to the user
Concordancers

- Take a text and create a **concordance** = display of words in context.
- Concordancers help learners understand how a given word is used.
  - For example, is the word *data* in English singular or plural?

  contract to supply voice and data communications within the Tunnel in
giving control over how much data is sent over the network
humanists to fit their special data to the software, rather
27 mm. But these data are for fourth-year crabs.
Morphological analysis

- **Lemmatization** = extract the lemma, or stem, of a word. (e.g. lemma of *corpora* is *corpus*.)
- **Morphological generation** = generate different forms of a word based on its lemma and part of speech

These processes are used to:

- help provide drill material for learners
- facilitate dictionary lookup (which can be very difficult otherwise for highly-inflected languages)

GLOSSER, for example, is a system that uses morphological processing to speed up dictionary look-up (100 times faster) (Nerbonne 2003)
**Authentic Text ICALL** attempts to connect learners to appropriate naturally-occurring texts

- Allows students to find examples in target language related to their interests
- Allows for more exploration and something akin to "immersion"

There are challenges in:

- Extracting appropriate exercises from the real texts
- Finding level-appropriate texts
The WERTi System
Working with English Real-Texts: An Intelligent Workbook for English

WERTi is an “intelligent automatic workbook, providing an unlimited number of activities designed to foster awareness of English grammatical forms and functions”

▶ Learners select a topic which fits their interests
▶ Webpages are returned, which learners interact to learn about, e.g., prepositions
  ▶ Learners can choose to see prepositions in color; click on them; or fill in blanks

Crucially, the exercises are *generated* on the fly

▶ Pre-existing NLP technology (e.g., a POS tagger) is used to spot the relevant categories
The REAP Project
Reader-Specific Lexical Practice for Improved Reading Comprehension

In the REAP system:

- Teachers have target vocabulary items
- REAP finds appropriate texts for learners, based on their individual profile
  - Learners get individualized vocabulary practice from authentic web texts

There are several challenges in extracting text for reading

- Each extracted text is analyzed for its “syntactic features, readability, length, and the occurrence of target vocabulary”
- Information retrieval and statistical NLP techniques are used to find appropriate texts
Parser-Based (I)CALL

Parser-Based ICALL systems generally fall along the following lines:

- System presents the learner with an exercise
- Learner inputs an answer, possibly with errors, i.e., a potentially **ill-formed** sentence
- The parser processes this sentence
  - Identifying where, if at all, it was incorrect
  - The nature of the error
- Feedback is then presented to the student

We’ll look at three different example systems:

- e-Tutor (German Tutor): Heift & Nicholson
- BANZAI/Robo-Sensei: Nagata
- TAGARELA: Amaral & Meurers
e-Tutor (German Tutor)

e-Tutor (Heift & Nicholson 2001) is used at Simon Fraser University to teach German to students; it is:

- general, i.e., allows for any native language (L1)
- able to capture different kinds of errors ... because the exercises are very constrained

Student input is put through the following modules and stops with feedback when the first error is encountered

1. String match: if the input matches a pre-defined correct answer, we know it’s good.
   - Prevents time-consuming analysis for perfect answers
2. Punctuation check: is any punctuation missing?
More on system architecture

3. Spell check: run an off-the-shelf spell checker on the input and get the **lemmas**
   - Idea: eliminate the really basic errors.
   - Problem: sometimes a “misspelled” word is a sign of lack of grammatical competence, e.g. *runned*

4. Example check: are the right words being used?
5. Missing word check: are any words missing?
6. Extra word check: are any words added?
   - These 3 steps (example, missing word, and extra word checks) all are based on the notion that the exercise has *pre-defined* all the acceptable words
More on system architecture (cont.)

7. Word order check: match the user word order with the correct word order

8. Grammar check
   ▶ This is the most complicated part of the process, the one which requires linguistic knowledge (syntax)
   ▶ About 60% of errors make it to this stage.

9. Catch-all: just in case everything else fails

Note:
   ▶ Heift’s system works so well because the exercises themselves are constrained, as we will see
   ▶ The approach is very **modular** = each check is an independent program
Kinds of exercises

Here are some example exercises from Heift’s system, outlined in http://zif.spz.tu-darmstadt.de/jg-06-2/beitrag/heift2.htm

- Dictation
- Build a Phrase
- Which Word is Different
- Word Order Practice
- Fill-in-the-Blank
- Build a Sentence
Dictation

Student hears a sentence in German and types it in. They are told if they are correct, and if not, why.
Dictation (cont.)

Pros:
- Input is very constrained.
- Very useful to be able to practice listening by oneself.
- Won’t take up class time.

Cons:
- Requires multimedia resources & may take awhile to prepare.
- Not very contextualized.

Such good and bad points could be given for all of the following exercises.
Build a Phrase

Build up a complete phrase (e.g., a noun phrase) based on a given picture—in other words, provide your own vocab.
Which Word is Different

Given 4 words, pick the one which differs from the others.

![Image of a language learning activity](image-url)
Word Order Practice

Take all the given words and arrange them into a sentence.

Guten Tag, Trude!
Ziehen Sie die folgenden Wörter in das Textfeld.

Übung 5 von 15

einem oder Wohnung? einer Haus

Wohnst du in

Prüfen Lösung Neu laden Weiter >>
Fill-in-the-Blank

Give the learner a lemma or choice of lemmas, and they have to fill in the blank.

![Fill-in-the-Blank Example](image)
Build a Sentence

Use all the given words (lemmas) and create a grammatical German sentence.

Advanced learner output here: “There is an error in gender with the subject.”
Robo-Sensei is proprietary software developed for teaching introductory Japanese courses

- Exercises cover a range of Japanese constructions
  - Extensive metalinguistic feedback is given to the learner
  - Noriko Nagata has shown, to some extent, that more difficult constructions are better served with this kind of feedback
- Hand-written rules cover Japanese morphology and syntax patterns
  - Comparison is done between the analysis of a learner’s input and a target analysis, in order to spot the error(s) and provide effective feedback
TAGARELA is a system for individualized instruction of Portuguese at Ohio State

- It features standard exercises, as found in foreign language workbooks
- NLP processing is used to detect spelling, morphological, syntactic, and semantic errors
- A student model is kept to track performance and to choose appropriate feedback
  - An instruction model allows the instructor to state what is important
TAGARELA system overview

Expert Model
- Segmentation model of language
- Tokenizer
- Non-word spell check
- Shallow Parser
- Deep Processing
- Lexical look-up/Morpho analysis
- Lexicon

Content modules
- correct answer
- required words
- required stems
- WN module

INPUT

Analysis Manager
- Form Analysis
- Content Analysis

Instruction Model
- Activity Info
  - Exercise requirements
  - Feedback requirements
- Error Info
  - Error Taxonomy (E.T.)
  - E.T. and feedback

Student Model
- Student info
  - Name, gender, level, etc
- Grammatical competence
  - Performance by knowledge units
- Strategic competence
  - Performance by task environment

Output

Feedback Manager (pedagogical modules)
- Feedback Generation
- Error Filtering
- Error Taxonomy (E.T.)
- E.T. and feedback
- Student analysis
- Feedback selection
- Ranking

ANOTATED INPUT

Feedback Manager (pedagogical modules)
- Error Filtering
- Error Taxonomy (E.T.)
- E.T. and feedback
- Student analysis
- Feedback selection
- Ranking

OUTPUT
Demand-driven architecture

Different from the e-Tutor, TAGARELA works in a demand-driven fashion; the analysis manager:

- receives input from the student
- gather the necessary information from:
  - instruction model
  - student model
- decides on the best processing strategy
  - which NLP modules to call
  - in which order (as opposed to linearly)
- calls NLP modules to process input, producing an input annotated with linguistic properties
- hands the annotated input to the feedback manager
The parser-based systems we’ve just seen have to deal with errors in learner input:

- How do we adapt our technologies to find errors?
- Do we tailor the system to a particular kind of learner?
- Do we tailor the system for an individual learner?
- What is the exact error?
- How many errors are there?
How do we adapt technology to find errors?

Our parsers, morphological analyzers, and so on, are made to handle well-formed input.

- Use so-called **mal-rules** = rules which are added to your grammar that handle error cases.
  - e.g., A singular noun and a plural verb are allowed to combine, but it is marked as an error.
  - \( S_{\text{error}} \rightarrow \text{NP}_{sg} \text{ VP}_{pl} \)

- Modify your technology: a parser can be reworked to handle ill-formed input.
  - Parsers normally just “die” when handling bad input.
  - e.g., I’ll parse *John are big*, but I’ll tell you that I didn’t like it and where it went wrong.
Do we tailor the system for an individual?

Do we keep track of a **student model** = what level each student is at, for a given task?

- Allows us to say which grammatical points need more work for a given individual.
- Allows us to give different feedback based on the learner’s abilities.
- Make sure the learner knows the terminology presented in the feedback, e.g., for the input *John are big*.
  - Beginner: “*John* is a singular subject and *are* is a plural verb.”
  - Intermediate: “There is a subject-verb agreement error.”
  - Advanced: “There is an error.”
Constraining the domain

Systems like the e-Tutor work well because they constrain what it is that the students can talk about.

- Input is fairly free, but students select from a pool of vocab and grammatical structures.
- Students are fairly introductory, so no exercise is too complex.
  - Compare teaching people subject-verb agreement vs. teaching them counterfactual clauses in English.
- All the pre-processing steps (before the grammar check) allow the grammar checker to assume well-formed input.