First language acquisition

How do babies learn language?

- Lack of explicit instruction?
- Positive evidence of language: children learn language based on all the possible sentence around them.
- Parents generally correct content, not grammar.
- Motherese ("baby talk"): different lexicon, intonation, topics, turn-taking.
- Tune in on relevant distinctions: synapses still taking shape in the brain.

Babies typically follow the same general stages of learning:

- Babbling at 6 months
- First words around 11-12 months (able to comprehend more than able to produce)
- Two-word stage around 18-24 months
- Utterances of varied length, vocab increasing, etc.

Second language learning differs from first language acquisition:

- Explicit knowledge of a language
- Consciously learning process
- Formal teaching helps

Language-specific stages of learning

- It can also help to identify when learners typically pick up specific constructions.

English learners, for example, typically follow this pattern:

1. Progressive -ing, plurals, the verb to be
2. Auxiliary verbs, articles (a, an, the)
3. Irregular past tense
4. Regular past tense, third person singular -s, possessive -s

We have already mentioned implicit vs. explicit instruction.

The best balance of the two for L2 learners is still being debated.

The two extremes are:

- "Drill-and-kill": repetition of exercises
- Immersion: all the learner needs is well-formed input

Adult learners are susceptible to transfer errors = something from L1 interfering with L2.

E.g., East Asian speakers typically mix up a and the in English: no such distinction in their language.
**Where does computer-aided language learning (CALL) fit in?**
- Wherever foreign language teaching is unavailable, inconvenient, or unfordable.
- CALL can be used in different setups, in particular:
  - self-study
  - supplement to in-class learning
- CALL is a big business: 106 million Euro (about $120 million) spent on CALL products in Europe in 1994. US market is twice as big. (Nerbonne 2003)

**Why CALL?**
- Frame-based systems
  - Linear systems
  - Branching systems
  - Concordancers
  - Finding Errors and Providing Feedback
  - User and task specific feedback
  - Example System
  - System architecture
  - Feedback

**Frame-based systems**
- **Frame-based systems** match student answers with a set of correct and incorrect answers stored in a frame.
- One can distinguish several types:
  - linear systems
  - branching systems
  - generative systems
- Typical for early CALL systems.

**Linear systems**
1. pose a question
2. accept an answer
3. inform 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.

**Generative systems**
- Generate new questions each time system is used.
  - Usually don’t have the same session twice
  - Based on some **algorithm** = sequence of commands, in this case used to generate new questions

**Problems with frame-based systems**
- No deep understanding of question domain.
- Merely match answers with questions.
⇒ Could be more than one right answer, as with a translation task.

**Example System**
- System architecture
- Feedback

**Problems with frame-based systems**
- Correct response: stay on the same layer → ask the next question in that linear system.
- Incorrect response → system jumps (or branches) to the question in the layer below to which the current question is linked.
  - If the question in the lower level is answered correctly, the system returns to the higher level.
  - If the lower-level question was answered wrongly, a jump is made to a yet lower level offering.
⇒ Much more arduous to come up with the question sets.
What is a correct answer to an exercise?

Take this fill-in-the-blank exercise (Heift):

- Today is November 10. What date is tomorrow?
  - Tomorrow is __________
    - The eleventh. [Correct]
    - November 11. [Incorrect?]
  => Have to allow for multiple right answers.

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?
  => for all right answers.
  contract to supply voice and giving control over how much humanists to fit their special data communications within the Tunnel in data communications within the Tunnel in data is sent over the network data is sent over the network data are for fourth-year crabs .

Text alignment

Show learners texts which are aligned between two languages.

- Advanced learners might benefit by seeing how word usage in their native language correspond to word usage in L2.
- Beginning learners would be overwhelmed.

Morphological analysis

- Lemmatization = extract the lemma, or stem, of a word. (e.g. lemma of running is run; lemma of corpora is corpus.)
- Morphological generation = generate different forms of a word based on its lemma and part of speech, or word class.
  => help provide drill material for learners
  => facilitate dictionary lookup (which can be very difficult otherwise for "highly inflected" languages – e.g. the lemma of Russian berut is brat)
GLOSSER, for example, is a system that uses morphological processing to speed up dictionary look-up (100 times faster) (Nerbonne 2003)

Syntactic analysis

- Syntactic generation = system creates sentences based on lemmas/words
- Syntactic parsing = clarify linguistic structure
- Syntactic processing allows us to deal with unrestricted learner input, such as essays, and give sensitive feedback for errors.
- Morphological and syntactic processing can also help make the learner more aware of what language is made up of.
- Meta-linguistic knowledge = knowledge about language.

Problems of Syntactic Processing

Main problem with syntactic processing: too many analyses. I saw the mouse in the house by the garden.

- by the garden can modify saw, mouse, or house.
- When learners type inconsistent sentences, you may have to allow for even more analyses.

Intelligent Computer-Aided Language Learning (ICALL)

- Intelligent CALL focuses on using linguistics and natural language processing to make CALL better.
- Examples include:
  - Concordancers
  - Text alignment
  - Speech recognition and synthesis
  - Morphological processing
  - Syntactic processing

Speech recognition and synthesis

- ASR: check and improve pronunciation.
- TTS: generate pronunciations of isolated words.
  If you’re using a paper dictionary, you have to base your pronunciations on a phonetic transcription of some sort.
Types of systems for error recognition (and feedback)

- Heif (Intelligent Language Tutoring Systems for Grammar Practice)
  - systems which present only the correct answer → no attempt to find an error
  - systems which pinpoint the error by a letter-by-letter comparison between student's answer and correct answer (linear systems)
  - systems which anticipate wrong answers (mal-rules)
  - systems which use NLP and provide linguistic analysis of sentence

Error recognition issues

- 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?

Do we tailor the system to a particular kind of learner?

- Some systems write mal-rules specifically designed to handle a particular type of learner, e.g. Korean learners of English.
- Can look at corpora and find the most common errors → can create an error typology = a classification of errors into different groups.
- Main problem is a lack of generality.

Error analysis

- So, we have a lot of different technology we can use.
- For many of the exercises we will use, the user types in something, and, using some technology, we want to find the error(s) in it.
- Two main issues:
  - Error recognition: What is the error?
  - Feedback: What do you about the error? (What do you tell the learner?)

How do we adapt our technologies 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.
- 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 the linguistic reason why not.

What is the error?

- Take this hypothetical example of someone learning English: Swimmer liked to swim.
- Did the learner mean:
  - Swimmers liked to swim. (problem with plurals)
  - The/A swimmer liked to swim. (problem with determiners)
  - Finding an error is one thing. Figuring out what the learner meant is another (similar to spell checking).

Number of errors

- Heif 2001 reports that 40% of sentences for German learners contained more than one error.
  - Don't want to overwhelm students with too much feedback.
  - Can present one error at a time. Instructors can divide the errors into primary errors and secondary errors – or rank them somehow.

An example system

Example system from Heif and Nicholson (2001), which is general (any native language) and which is able to capture different kinds of errors ... because the exercises are very constrained (as we will see later)

Student Input →
- String match: if the input matches a pre-defined correct answer, we know it's good.
  - Prevents time-consuming analysis for perfect answers.
- Punctuation check
More on system architecture

- Spell check: run an off-the-shelf spell checker on the input and get the **lemmas** = baseforms of words for the next step.
- Idea: eliminate the really basic errors.
- Problem: sometimes a "misspelled" word is a sign of lack of grammatical competence, e.g. runned is "misspelled", but it might show a lack of knowledge about the English past tense.
- Example check
- Missing word check
- Extra word check

These 3 steps (example, missing word, and extra word checks) all are based on the notion that the exercise has **pre-defined** all the words which are acceptable for this answer.

Feedback (cont.)

Things to keep in mind when designing a system (somewhat obvious):

- Feedback needs to be accurate.
- Displaying more than one error message at a time is not helpful.
- Explanations should be short.

Kinds of feedback

- **Explicit correction** = explicitly giving the correct form, indicating that this is a correction.
- No, not Yo habla. You want to say Yo hablo.
- **Recast** = reformulating all or part of the student's utterance, without the error, and not indicating that this is a correction.
- STUDENT: Yo habla español. TEACHER: Yo hablo español tambien. (I speak Spanish also.)
- **Clarification request** = asking for a clarification.
- What? What did you mean?

Kinds of exercises

Here are some example exercises from a German system (Heift), outlined in [http://www.spz.tu-darmstadt.de/projekt_ejournal/ig_06_2/beitrag/teilt12.htm](http://www.spz.tu-darmstadt.de/projekt_ejournal/ig_06_2/beitrag/teilt12.htm)

- **Dictation**

Diction

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

```
Gestern trat die Nina ins Zimmer ein. (How should the sentence be?)

Nina trat ins Zimmer ein. (Correct)

```

Dictation (cont.)

Good points:

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

Bad points:

- Requires multimedia resources.
- Takes a long time to prepare.
Build a Phrase

Build up a complete phrase (e.g. NP, but not a whole sentence) based on a given picture (i.e. provide your own vocab).

Fill-in-the-Blank

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

Build a Sentence

Use all the given words (lemmas) and create a grammatical German sentence. Advanced learner output (“There is an error in gender with the subject.”):

Constraining the Domain

As we said before, Heil’s system works pretty well because she constrains what it is that the students can talk about.

Demos

Tools:

- Concordancer
- Morphological Analyzers XEROX online Demo (English)
- Parser
- ICALL demos:
  - GLOSSER