Handout 1: What is Computational Linguistics?

1. Main areas

- **a.** Human language technology (HLT) – MT, Speech, IE
- **b.** Natural language processing (NLP) – AI, conversational agents
- **c.** Computational psycholinguistics
- **d.** Digital linguistics

Language technology

2. Conversation systems: Siri, dial-up spoken language systems, game AIs, chatterbots, HAL.

3. Speech recognition (ASR), speech synthesis (TTS), spelling correction, optical character recognition (OCR)

4. Information retrieval, web search

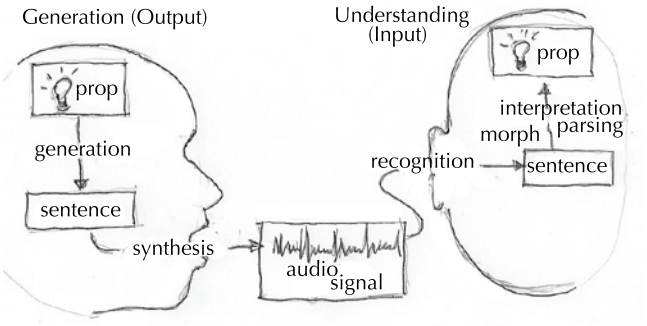
- **a.** Text classification: speaker identification, language identification, author identification, gender identification
- **b.** Information extraction: Google, biomed, NLP enabling research in other fields
- **c.** Question answering: Watson

5. Machine translation, voice-to-voice translation

6. Misc: computer-aided language learning (CALL), tutoring, natural language generation for weather reports, grammar checkers, hyphenators

Conversational agent

7. Schematic



8. At heart, language is a transducer
   a. NL generation: proposition → sentence
   b. NL understanding: proposition ← sentence
   c. A sentence – that's clear:

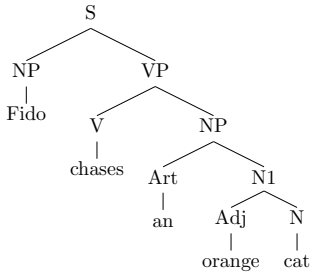
      Fido chases an orange cat

   d. So what is a meaning? Higher-order predicate calculus:

      $$\text{EXISTS}(\lambda c : \text{CAT}(c) \wedge \text{ORANGE}(c) \wedge \text{CHASES}(\text{FIDO}, c))$$

   e. But what is the meaning of *that*? Shared environment provides the touchstone for shared meaning.

9. Proposition and sentence structure
   a. Compute the meaning of "orange cat" $\lambda x : O(x) \wedge C(x)$ from meaning of "orange" O and meaning of "cat" C.
   b. The **syntactic structure** (or **parse tree**) gives the sequence of compositions by which the value is computed:



10. Output side: **natural language generation**
    a. **Utterance planning** (Intention): deciding what to say. Output is a **proposition**.
    b. **Lexical choice:** predicates → English words
    c. **Syntactic realization:** map semantic relations onto English sentence structure
    d. **Morphological realization:** add inflection
    e. (Text-to-speech) **synthesis** (TTS): turn the sentence into an audio signal.

11. Input side: **natural language understanding**
    a. Automatic **speech recognition** (ASR): audio signal to phones (letters)
    b. Optical character recognition (OCR): image to letters

c. **Morphological analysis** and lexical look-up: letters to words

d. **Parsing:** words to syntactic trees

e. **Semantic interpretation:** syntactic trees to propositions

f. **Pragmatics:** going from literal meaning to intended message

g. **Discourse:** filling in unexpressed bits from conversational history (pronouns, definite expressions, ellipsis)

h. Reasoning & planning, database storage

12. After Russell & Norvig, p. 796:

```
def naive_communicating_agent (self, percept):
    self.update_state(percept)
    words = percept.speech_part()

    if words:
        tree = parse(words)
        lf = semantics(tree)
        readings = pragmatics(lf)
        meaning = disambiguate(readings)

        if meaning.type == 'command':
            # return value is next action
            # i.e., we obey the command
            return meaning.contents

        elif meaning.type == 'question':
            answer = self.kb.query(meaning.contents)
            return Say(description(answer))

        elif meaning.type == 'statement':
            # we believe anything we're told
            self.kb.store(meaning.contents)

    return planner(self.kb, self.state).first()
```

History

13. 1950s: Machine translation

a. Information theory & cybernetics: Weaver memorandum

> When I look at an article in Russian, I say: "This is really written in English, but it has been coded in some strange symbols. I will now proceed to decode."

b. Markov models, sentences as strings

c. Neural networks and complex systems. Self-regulation (control) reduces to communication among components (cybernetics).

d. Computational linguistics did not emerge from linguistics, but from machine translation

14. 1960s: Artificial intelligence

a. Eliza

b. Markov models rejected as inadequate, logic and grammar became model

c. Blocks world, story understanding

d. Database front-ends, e.g. LUNAR

15. Why is NLP so hard?

a. HAL become operational on January 12, 1992

b. Problem for symbolic approach: dealing with ambiguity

c. How many interpretations are there for "I made her duck"?

16. Kinds of ambiguity

a. Part of speech: *duck, her*

b. Syntactic: *make* transitive, ditransitive, infinitival complement

c. Word sense: *make* = cook, create, cause, transform; *duck* = animal, meat, toy, head-lowering

d. Acoustic: *aye mater duck!* wɪlijlzəbejəgaɪ

17. The need for robustness

a. "the are is enough"

b. New domains

> Reactive oxygen intermediate-dependent NF-kappaB activation by interleukin-1beta requires 5-lipoxygenase or NADPH oxidase activity.

c. Sapir: "all grammars leak"

18. 1990s: Introduction of machine-learning methods

a. Rebuilding MT from ground up

b. Ongoing synthesis of Markovian and grammatical methods, but incomplete

c. Modern CL is applied machine learning

This course

19. Syllabus

20. Will use the Natural Language Toolkit (NLTK)

- **a.** A Python library module
- **b.** We'll learn Python as we go.

21. Programming

- **a.** At least a little prior experience with programming is assumed
- **b.** But the course is explicitly designed to be accessible to students in linguistics, cognitive science, etc.

22. Syntax and semantics

- **a.** I'll assume you are familiar with basic parts of speech and can identify subject and predicate, prepositional phrases, and the like
- **b.** Linguistics students should take Ling 315 and 316 first

23. What we'll cover: see syllabus

Getting started

24. See "Start" link on CTools

- **a.** Need access to Python 3 and NLTK
- **b.** I recommend installing Anaconda (see "Start" link)
- **c.** Note: must be Python 3, not Python 2

25. Alternative: pre-installed on `chukar.dsc.umich.edu`

- **a.** Use SSH ("Secure SHell") to connect to chukar. Most machines have an SSH client installed.
- **b.** Macs: use `ssh` in Terminal window
- **c.** Gives you a Unix shell on chukar
- **d.** Set up environment: `~abney/cl/start`
- **e.** You can insert it in your `.login` file:

```
$ cat ~abney/cl/start >> .login
```

- **f.** Start python and load NLTK

```
$ python
Python 3.3.2 (default, Mar 20 2014, 20:25:51)
>>> from nltk.book import *
```

- **g.** Getting back out: c-D c-D exit

h. You'll need to know some basic Unix commands: `ls`, `pwd`, `cd`, `mkdir`, `cp`, `rm`, `cat`. See "Start" link.

26. Plain-text editor

a. `vim` or `emacs`. `pico` is also OK.

b. If you use an IDE that handles Python files, that's fine, too.

c. Open one window running python, one window running editor.

27. Loading a python file

a. Homeworks will ask you to set variables to indicate answers

b. E.g., "Compute the sum of two and two and store it in the variable `x`." In file `hw.py`:

```
x = 2 + 2
print('x=', x)
```

(The print is just so you can see what's going on.)

c. In python:

```
>>> import hw
x= 4
>>> hw.x
4
```

d. Alternatively, at Unix prompt:

```
$ python hw.py
x= 4
```

28. Suppose the next question asks you to set y to x times 3. Edit `hw.py`, then do

```
>>> from imp import reload
>>> reload(hw)
>>> hw.y
12
```

29. Transferring files between chukar and your machine

a. Use `http://mfile.umich.edu/`

b. Or use "M:" drive on Campus Computing Sites machine

c. Or, on a Mac, use `scp` in Terminal