Lisa Pearl
LING200, Summer Session I, 2004

Computational Linguistics – The Interface of Computers and Language

I. Computational Linguistics

A. Fairly recent field of study – only in the last few decades. Made big leaps and bounds since the Internet came about for big-time public use. (For instance, you're making use of computational linguistics every time you google for something.)

B. The big question: What does a computer need to know to analyze, understand, and even create sentences, paragraphs, and conversations or essays?
Ex: "This argument reads well but doesn't make a lot of sense when you think about it."

a. **grammatical knowledge**: "reads" = present tense, singular agreement, "doesn't" = inflection + negation, "this" = modifier of argument, etc…

b. **real world knowledge**: arguments are more likely to *be* read than do the reading, "make a lot of sense" = idiom meaning "seem sensible", etc….

c. **pronunciation knowledge**: reads = [ridz] and not [rɛdz], etc…

II. Computational Syntax

A. **Grammar** → think of as a set of rules which define a language

B. **Parser**: the machine or engine that applies these rules to make sense of input.

a. But what happens when there's more than one option for parsing a sentence, particularly when it's only partially completed?

b. "I saw her…"

i. "…in the ballroom."
(her = direct object)

ii. "…dance with that dashing prince."
(her = modifier of dance OR subject of an embedded clause)

c. What can a parser do at an ambiguous word?

i. **deterministic parsing**: pick one option (using a particular strategy, for instance) and stick with it until it turns out to be wrong.

1. pro: doesn't use a lot of resources
2. con: could be wrong, and then have to do a lot of clean-up work and backtracking

ii. **nondeterministic parsing**: keep multiple options open until enough information is obtained to rule one out.

1. pro: no need to clean-up a mess made by a wrong choice
2. con: could be expensive to keep more than one analysis option open

C. Top-Down vs. Bottom-Up Parsing - two ways to think about building a sentence

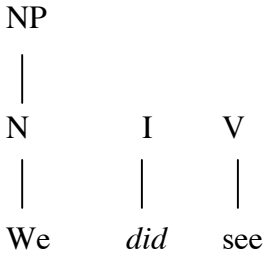
i. Bottom-Up

1. Start with Phrase Structure Rules (XP → …)

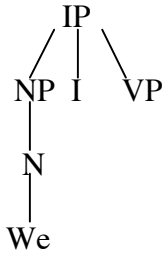
Ex: IP → NP I VP
NP → (Det) (AP) N (PP)
VP → V (NP) (PP)
PP → P NP

2. Each rule consists of a phrase (ex: IP) and what its components or **nodes** are (ex: NP I VP). Each node may be a **terminal node** (ex: I) or it may be a phrase itself (ex: NP) which can be expanded (aka a **nonterminal** node).

3. If you think of the phrase structure rules as forming the "tree" for the sentence, bottom-up parsing begins with the "leaves" – the terminal nodes – and fills them in with words. Then, when enough leaves have been filled to make a phrase, the phrase is built.



ii. Top-Down
1. A top-down parser infers structure as it goes along, whether or not the words to fill in that structure have been seen yet. So, as soon as the first word of the sentence is seen, it will infer the IP components.



2. This continues until the entire sentence is filled in.

III. Applications of Computational Linguistics

A. **Indexing & Concordances**

a. **indexing**: finding, identifying, and usually counting all occurrences of a certain word in a large text.
   i. Useful for determining frequencies of individual words in a particular document or over a **corpus** (set of documents) which represents some "general usage" of the English language.
   ii. Example: Francis & Kucera index of one million word corpus
   most frequent words: the, of, and, to, at, in , that, is, was, he

b. **concordance**: tells which words occur near other words.
   i. Example: a program of concordances might find that "flounces" occurs far more often with "she" than with "he" – which reflects something about the meaning that "flounce" has. (Perceived as a more feminine action.)
   ii. Computational Parsing Aspect: Useful for determining lexically ambiguous words. "ground" appearing in a document with "grass", "trees", "flowers" vs. "ground" appearing in a document with "grind".

c. Difficulties
    i. Determining part of speech of a word (ex: "dance" → context may or may not help), requires syntactic knowledge
    ii. Determining that forms are related (ex: "gone" and "went" are really both forms of "go"), requires morphological knowledge
d. Real Life Usefulness

Example: Suppose you're searching for information on the1986 movie *Labyrinth*. You type "labyrinth" into google and get the following as the first three hits:

Labyrinth Home Page: The **Labyrinth**: Resources for Medieval Studies
Labyrinth Latin Library: Armarium Labyrinthi: **Labyrinth** Latin Bookcase
The Grey Labyrinth: The Grey **Labyrinth** is a collection of puzzles, riddles, and paradoxes designed to stimulate lateral thinking.

You realize that there are many more "Labyrinths" out there than the 1986 movie. So what do you do? Try concordance information – add a term to the **search query**. Instead of "Labyrinth", type "Labyrinth movie". Immediately, you end up with the following:

Think Labyrinth: The Movie!: **...** I've acquired a good collection of **Labyrinth movie** memorabilia over the years. **...**

B. **Information Access & Retrieval**
a. When you google for a particular piece of information, how often do you come up with irrelevant web pages? ….probably a lot more often than you'd like. Or what about those websites that have a "type in your question here and search through our website"? They have a really hard time pulling up a relevant answer to whatever question you type in.
b. Information access & retrieval: goal is to be able to use linguistic information to pull out exactly the information that you want.
c. Why this is hard.
    i. Computers are good at sorting through lots and lots of information very quickly.
    ii. Computers *only* do what you tell them to do – they have no built-in knowledge of language.
d. How does google go about finding "relevant" documents, based on your query?
Example: "Where can I buy the movie Labyrinth?"
    i. Get rid of words which appear a lot (use frequency information) and which provide little information about what is relevant.
    "Where can I buy the movie Labyrinth?" → "buy movie Labyrinth"
    ii. Word frequency and **co occurrence** information are calculated and compared against the same information in each document google sorts through.
    Document 1 → "buy+movie+Labyrinth" High or Low?
    Document 2 → "buy+movie+Labyrinth" High or Low?
    Document 3 → "buy+movie+Labyrinth" High or Low?
    …

iii. Rank them according to how well the concordance information in the query matches the concordance information in the document (use fancy statistical measures) → put highest ranking documents out first.

Google #1 hit for "Where can I buy the movie Labyrinth?"
Amazon.com: DVD: Labyrinth (1986)

e. Other methods which can be used – shallow parsing to group NPs together. Searching by the heads of the NPs can also help find relevant documents.

C. **Machine Translation**

1. *Purpose*: take any written or spoken text from one language and transform it into another language.
2. *Problem*: Really, really hard. Why? Languages are not simply word-for-word translations of each other – they use different syntactic and morphological options.

Ex: Using worldlingo.com's translation tool, you can translate from English to Japanese and back again…with interesting results:
"*A slayer* with family and friends – **that sure as hell *wasn't in the brochure***." →
(Japanese) →
"The friend and the family **which *are* not truly *in*** *the murder person* ***pamphlet*** which it has."

Words can often be ambiguous, and then the computer must decide which sense of a word is appropriate.

Ex: "Surprisingly, the fairy penguins *smelled* quite awful."
smell = to use nose to sniff *or* to emit an odor?

Ex: using google's translation tool, you can translate from English to German and back.
"I leave bite marks" → (German) → "I leave bite *characters*."

3. *Why automatic translation has such a big market:* Billions of documents need to be translated every day. Take google as an example. When the information you want is found in a document in a foreign language, google often offers you the option of translating that foreign document into English.

D. **Automatic Summarization**

1. The process of automatic analysis of either a single article or a set of articles & the creation of an abstract reflecting the key ideas in a concise and coherent way.
2. *Why is it hard?* The computer must not only parse sentences but also extract *semantic content* from them. Requires some semantic representation of the information. In addition, **generation** of fluid, concise sentences to produce the abstract is also quite difficult.
3. *Why do we want it?* In an age of information overload, wouldn't it be nice if something could automatically extract the Cliff Notes version of everything for you?
4. *How do we go about doing it currently?*

a. frequent, informative words are extracted and glommed together. (Tends to produce crude and choppy sentences.)

b. starting to have sophisticated techniques to extract semantic content and sophisticated techniques of generation.

D. Speech Recognition & Speech Synthesis

1. A **speech recognition** system: take spoken language as input and "understand" it.
   a. Humans do this naturally and are very good at it – but it's actually quite hard.
   b. People have different accents and different pronunciations of sounds (free variation) - especially during rapid speech - which a computer must decipher into phonemes, glom together to produce words, parse into sentences, and extract semantic content from.
   c. One important application is for the physically disabled. Ex: Having a computer which can interpret the speech of a blind person so they can type on the computer.
   d. *How does it work*? Often use statistical data – for instance, when finding word boundaries.

3. A **speech synthesis** system: take written language and turn it into spoken language.
   a. Hard to make it sound natural. (Think of those automated services which say, "Press…1….now…")
   b. Practicality: helping the disabled. Ex: Reading to a blind person.
   c. More Practicality: Voice-activated database query – system reads out information you want over the telephone.

Exercises

1. Smelly Penguins.
"The penguins smell."

a. Give two meanings this sentence could have.
b. Draw the phrase structure (feel free to leave out X' categories).

An automatic translation system (worldlingo.com) has been used to translate this sentence into Spanish and back. It produced the following:
"The smell of the penguins"

c. How must this translation system have parsed the original phrase?
d. What punctuation is missing from the original system that would have produced this meaning?

2. Parser Woes

Suppose a parser uses a top-down processing strategy and assumes, until proven otherwise, that the phrases it parses will all be IPs and will all begin with an NP subject. Suppose it is then given the following sentence to parse:

Watch dogs bark.

a. Draw the phrase structure it would assign to this sentence. (Leave out X' categories.)
b. What meaning would the sentence with that phrase structure have?
c. What other interpretation could this sentence have?
d. What sort of parsing is this parser using to determine the structure of the sentence?

3. Word Dissociation
(*Thieved and slightly altered from http://www.ravenblack.net/games/worddissociation.html.*)
"Word dissociation is a game you can play on your own, at work if you have internet access and don't mind getting fired. The idea of the game is to find two common words that are totally dissociated. For the purpose of the game, a word is considered common if a search on the websearch engine you are to use produces 10000+ hits. A pair of words is considered dissociated if a search on the same search engine for a page including both words produces 0 hits. If all of these conditions, you score 1 point."

a. On its own, a word from a dissociated pair would be found to have what kind of frequency by an indexing program?
b. What kind of concordance ranking would a dissociated pair of words have?

4. Retrieval Cleverness
What words would a clever program get rid of from this query before searching through all the documents in its database? Why do you think it would get rid of these words?

Where can I find the absolute best place to buy large quantities of body glitter?

5. Machine Translation Blues
Using worldlingo.com's automatic translation program took sentence 1 in as English input, translated it into Korean, and gave sentence 2 as English output:

1. **"I'm mad, you're mad – we're all mad here."**
2. "**Me it goes mad, It spreads out and it goes mad and - we all is here it goes mad**."

a. What does this tell you about how similar Korean and English are in terms of structure?

The same original sentence (sentence 1), cycled through worldlingo.com in Spanish and then back again, produced sentence 3.

3. **"I am angered, you are angered - we are all angry here."**

b. What does this tell you about how similar English and Spanish are in terms of structure? Does this make you think English & Spanish are closer in structure than English & Korean?

6. Speech Recognition Snafus
Suppose a speech recognition system produces the following text:

"It's an old saying everyone knows. ... **Sloane's Teddy wins the race**."

a. What is it likely that the old saying is, rather than what the speech recognition system got?
b. Where did the speech recognition system go wrong?