Introduction to Advanced Natural Language Processing (NLP)

L645 / B659
Advanced NLP

Dept. of Linguistics, Indiana University
Fall 2015
Computational linguistics is the study of computer systems for understanding and generating natural language.

Simply put, computational linguistics is the scientific study of language from a computational perspective. Computational linguists are interested in providing computational models of various kinds of linguistic phenomena. These models may be "knowledge-based" ("hand-crafted") or "data-driven" ("statistical" or "empirical"). Work in computational linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon; and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system. . . .
Definition of CL 2 (cont.)

...Indeed, the work of computational linguists is incorporated into many working systems today, including speech recognition systems, text-to-speech synthesizers, automated voice response systems, web search engines, text editors, language instruction materials, to name just a few.

(http://www.aclweb.org/archive/misc/what.html, retrieved 8/3/15)
Short History

- 1950s: Machine Translation
- 1964: ALPAC Report
- 1966: ELIZA
- 1973: SHRDLU
- 1980s: knowledge-based CL
- 1990s & beyond: statistical / machine learning approaches in CL
Early 1950s

- Machine Translation (MT): one of the earliest applications of computers
  - Major players: US and USSR
  - Russian to English and reverse

- Georgetown University, Washington system:
  - Translated sample texts in 1954
  - Euphoria: a lot of funding, many groups in US, USSR
    - BUT: the system could not scale up
ALPAC Report

- Assessed research results of groups working on MTs
- Conclusions:
  - MT not possible in near future
  - Funding should cease for MT!
  - Basic research should be supported
- Word-by-word translation does not work
  - Linguistic knowledge is needed
ELIZA

- The first chatterbot – a computer program that mimics human conversation
  - Author: Joseph Weizenbaum (MIT)
- Simulation of a (Rogerian) therapist
  - User types in some statement or set of statements in natural language
  - ELIZA then analyzes the user’s statement and generates some response
- Basic technology: pattern matching

USER: You don’t argue with me.
ELIZA: WHY DO YOU THINK I DON’T ARGUE WITH YOU
Interaction with a robot in a block world.
  - Author: Terry Winograd (MIT)

The user can:
  - ask the robot to manipulate the blocks
  - ask it about the blocks configurations
  - ask it about its reasoning
  - update facts

“Understands” language in a limited domain by using syntactic parsing and semantic reasoning
  - Large scale grammar of English + parser
  - Procedural semantics for words and phrases
Knowledge-Based CL

- Proof of concept & manually-written rules
  - Linguistic/logic paradigm extensively pursued
  - Later: development of linguistic formalisms (Lexical Functional Grammar, Head-Driven Phrase Structure Grammar, Tree Adjoining Grammar, etc.)

- Limitations:
  - Not robust enough
  - Few applications
  - Not scalable ... though, systems are still getting better

Addressing the limitations led to the more recent statistical approaches
Statistical / Machine Learning Approaches

- Instead of writing rules, have computer learn rules / regularities
- Approach massive ambiguity problem by probabilities
- Need annotated data for training
  - Data sparseness problem
  - Unsupervised learning does not help: no linguistically relevant rules
To sum, two main approaches to doing work in NLP:

- **Theory-driven (≈ knowledge-based):** working from a theoretical framework, come up with a scheme for an NLP task
  - e.g., parse a sentence using a handwritten HPSG grammar
- **Data-driven (≈ statistical):** working from some data (and some framework), derive a scheme for an NLP task
  - e.g., parse a sentence using a grammar derived from a corpus

The difference is often a matter of degree

- This course is more data-driven & probabilistic
Consider the following (Abney 1996):

(1) The a are of I.
(2) John saw Mary.

- *The a are of I* is an acceptable noun phrase (NP): *a* and *I* are labels on a map, and *are* is measure of area

- *John saw Mary* is ambiguous between a sentence (S) and an NP: a type of saw (*a John saw*) which picks out the *Mary* we are talking about (cf. *Typhoid Mary*)

We don’t get these readings right away because they’re rare usages of these words

⇒ Rarity needs to be defined probabilistically
Probability

Wide-coverage of rules

Grammar rules work sometimes & not others

▶ Typically, if a noun is premodified by both an adjective and another noun, the adjective must precede the modifying noun

(3) tall (A) shoe (N) rack
(4) *shoe (N) tall (A) rack

▶ But not always:

(5) a Kleene-star (N) transitive (A) closure
(6) highland (N) igneous (A) formations

If language is categorical and you have a rule which allows N A N, then you have to do something to prevent shoe tall rack.
Using probabilities
The Ambiguity of Language

Language is ambiguous in a variety of ways:

- Word senses: e.g., *bank*
- Word categories: e.g., *can*
- Semantic scope: e.g., *All cats hate a dog.*
- Syntactic structure: e.g., *I shot the elephants in my pajamas.*

Often, however, of all the ambiguous choices, one is the best
(7) Our company is training workers
Less intuitive analyses (1)

```
S
  NP  VP
  Our company Aux NP
    is VP
      V NP
        training workers
```
Less intuitive analyses (2)

S
   /\  /
  NP  VP
 /   /
Our  V
  /   /
  is  Adj
   /   /
  training  NP
           /   /
           workers
Corpora

We can **induce** probabilistic information from language data, potentially data annotated with linguistic information.

- Thus, we will become familiar with processing large texts, i.e., **corpora**
- Corpora are often annotated with linguistic mark-up, such as part-of-speech labels or syntactic annotation

These corpora will serve as our data from which to learn probabilities

- Corpora are not the only lexical resources out there; dictionaries (e.g., WordNet) are also important, but these are often derived from corpora
Using corpora for simple analysis

Word counts

We can use corpora to give us some basic information about word occurrences

- Count **word types** = number of distinct words there are in the corpus
- Count **word tokens** = number of actual word occurrences in the corpus; multiple occurrences of the same word type are counted each time

If we compare word types and tokens, we see that there are:

- a few word types which occur a large number of times (often function words)
- a large number of word types which occur only a few times or only once
Zipf’s Law

This idea is formulated in **Zipf’s Law** = the frequency \( f \) of a word is inversely proportional to its rank \( r \)

\[
(8) \quad a. \ f r = k, \text{ where } k \text{ is some constant, or } f = \frac{k}{r} \text{ (Zipf)} \\
b. \ f = P(r + \rho)^{-B}, \text{ where } P, \rho, \text{ and } B \text{ are parameters which measure a text’s richness (Mandelbrot)}
\]

Mandelbrot adjusted Zipf’s Law to better handle high and low ranking words; with \( B = 1 \) and \( \rho = 0 \), it is identical to Zipf’s Law (where \( P = k \)).

- Important insight: most words are rare!
Linguistic levels

- phonetics / phonology
- morphology
- POS annotation
- syntax
- lexical semantics
- discourse
CL Analysis

- finite-state morphology (analysis + generation)
- POS tagging
- parsing
- word sense disambiguation
- detect selectional restrictions (kill, murder, assassinate)
- shallow inference (X killed Y ⇒ Y is dead)
- anaphora / coreference resolution
Concepts Borrowed from Computer Science

- finite-state automata / transducers
- search: divide and conquer, beam search, nondeterminism, guides and oracles
- parsing (compilers)
- dynamic programming
- machine learning approaches: decision trees, $k$-nearest neighbors, clustering, support vector machines, ...