Topics for today
- Introduction to computational morphology
- Basics of English morphology
- Finite-state morphological parsing

Why study NLP?
- Useful applications
- Interdisciplinary
- Challenging

Why is NLP hard?
Ambiguity!!! …at all levels of analysis 😊
- Phonetics and phonology
  - “I scream” vs. “ice cream”
- Morphology
  - unionized = union + ized? un + ionized?
- Syntax
  - Squad helps dog bite victim.
- Semantics
  - Jack invited Mary to the Halloween ball.
- Discourse
  - Merck & Co. formed a joint venture with Ache Group, of Brazil. It will be called Prodome Ltd.

Why is NLP hard?
Ambiguity!!! …at all levels of analysis 😊
- Pragmatics
  - Concerns how sentences are used in different situations and how use affects the interpretation of the sentence.
  - “I just came from New York.”
    - Would you like to go to New York today?
    - Would you like to go to Boston today?
    - Why do you seem so out of it?
    - Boy, you look tired.
Additional Course Info

- Time: Mondays and Wednesdays, 11:15-12:05
  - Occasional Fridays
- Office hours: Tuesday 3-4, Thursday 1-2
- Course Materials:
  - Lecture Notes, Readings, Assignments
  - Other Handouts
  - Lillian Lee’s list of on-line NLP resources

Syllabus (tentative)

- Introduction (1 lecture)
- History and state-of-the-art (1 lecture)
- Morphology (2 lectures)
- N-grams (1 lecture)
- Context-sensitive spelling correction (1 lecture)
- Part-of-speech tagging and HMMs (2 lectures)
- Parsing (3 lectures)
- Partial parsing (2 lectures)
- Semantic analysis (2 lectures)
- Inference and world knowledge (1 lecture)
- Information extraction (1 lecture)
- Lexical semantics and WSD (2 lectures)
- Discourse processing (3 lectures)
- Generation (2 lectures)
- Machine translation (1 lecture)

Reference Material

- Recommended text book:
- Other useful references:
  - Others listed on course web page…

Prereqs and Grading

- Prerequisites
  - Elementary computer science background, elementary knowledge of probability, familiarity with context-free grammars.
- Grading
  - 30%: critiques of selected readings and research papers
  - 60%: final project. Grade based on
    - (1) preliminary project proposal (3/12),
    - (2) project literature survey (4/9),
    - (3) project presentation (4/21-4/30),
    - (4) final write-up (5/14).
  - 10%: participation
Readings and Critiques

Critique Guidelines

- <=1 page, typed (single space)
- The purpose of a critique is **not** to summarize the paper; rather you should choose one or two points about the work that you found interesting.
- Examples of questions that you might address are:
  - What are the strengths and limitations of its approach?
  - Is the evaluation fair? Does it achieve it support the stated goals of the paper?
  - Does the method described seem mature enough to use in real applications? Why or why not? What applications seem particularly amenable to this approach?
  - What good ideas does the problem formulation, the solution, the approach or the research method contain that could be applied elsewhere?
  - What would be good follow-on projects and why?

Critique Guidelines

- Are the paper’s underlying assumptions valid?
- Did the paper provide a clear enough and detailed enough description of the proposed methods for you to be able to implement them? If not, where is additional clarification or detail needed?

- Avoid **unsupported** value judgments, like “I liked...” or “I disagreed with...” If you make judgments of this sort, explain why you liked or disagreed with the point you describe.
- Be sure to distinguish comments about the writing of the paper from comment about the technical content of the work.

Topics for Today

- Finish up general introduction
- More details on the course, course requirements, etc.
  - Student info sheet
  - Brief history of NLP
Early Roots: 1940’s and 1950’s

- Work on two foundational paradigms
  - Automaton
    - Turing’s (1936) model of algorithmic computation
    - Kleene’s (1951, 1956) finite automate and regular expressions
    - Shannon (1948) applied probabilistic models of discrete Markov processes to automata for language
    - Chomsky (1956)
    - First considered finite-state machines as a way to characterize a grammar
  - Led to the field of formal language theory

- Probabilistic or information-theoretic models for speech and language processing
  - Shannon: the “noisy channel” model
  - Shannon: borrowing of “entropy” from thermodynamics to measure the information content of a language

Two Camps: 1957-1970

- Symbolic paradigm
  - Chomsky
    - Formal language theory, generative syntax, parsing
    - Linguists and computer scientists
    - Earliest complete parsing systems
    - Zelig Harris, UPenn
    - We’ll look at this parser in a critique reading!!

- Artificial intelligence
  - Created in the summer of 1956
  - Two-month workshop at Dartmouth
  - Focus of the field initially was the work on reasoning and logic (Newell and Simon)
  - Early natural language systems were built
    - Worked in a single domain
    - Used pattern matching and keyword search
Two Camps: 1957-1970

- **Stochastic paradigm**
  - Took hold in statistics and EE
  - Late 50’s: applied Bayesian methods to OCR
  - Mosteller and Wallace (1964): applied Bayesian methods to the problem of authorship attribution for *The Federalist* papers.
  - Another critique reading!!!

Additional Developments

- **1960’s**
  - First serious testable psychological models of human language processing
    - Based on transformational grammar
  - First on-line corpora
    - The Brown corpus of American English
      - 1 million word collection
      - Samples from 500 written texts
      - Different genres (news, novels, non-fiction, academic,....)
      - Assembled at Brown University (1963-64, Kucera and Francis)
    - William Wang’s (1967) DOC (Dictionary on Computer)
      - On-line Chinese dialect dictionary

1970-1983

- **Explosion of research**
  - Stochastic paradigm
    - Developed speech recognition algorithms
      - HMM’s
      - Developed independently by Jelinek et al. at IBM and Baker at CMU
  - Logic-based paradigm
    - Prolog, definite-clause grammars (Pereira and Warren, 1980)
    - Functional grammar (Kay, 1979) and LFG

1970-1983

- **Explosion of research**
  - Natural language understanding
    - SHRDLU (Winograd, 1972)
    - The Yale School
      - Focused on human conceptual knowledge and memory organization
    - Logic-based LUNAR question-answering system (Woods, 1973)
  - Discourse modeling paradigm
Revival of Empiricism and FSM's

- 1983-1993
  - Finite-state models
    » Phonology and morphology (Kaplan and Kay, 1981)
    » Syntax (Church, 1980)
  - Return of empiricism
    » Rise of probabilistic models in speech and language processing
    » Largely influenced by work in speech recognition at IBM
  - Considerable work on natural language generation

A Reunion of a Sort...

- 1994-1999
  - Probabilistic and data-driven models had become quite standard
  - Increases in speed and memory of computers allowed commercial exploitation of speech and language processing
    » Spelling and grammar checking
  - Rise of the Web emphasized the need for language-based information retrieval and information extraction

Statistical and Machine Learning Approaches Rule!

- 1992 ACL: 24% (8/34)
- 1994 ACL: 35% (14/40)
- 1996 ACL: 39% (16/41)
- 1999 ACL: 60% (41/69)
- 2001 NAACL: 87% (27/31)

<table>
<thead>
<tr>
<th>Year</th>
<th># of papers</th>
<th>no ML</th>
<th>some ML</th>
</tr>
</thead>
<tbody>
<tr>
<td>1992</td>
<td>24% (8/34)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1994</td>
<td>35% (14/40)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1996</td>
<td>39% (16/41)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1999</td>
<td>60% (41/69)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2001</td>
<td>87% (27/31)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

WVLC and EMNLP Conferences

- Workshop on Very Large Corpora
- Conference on Empirical Methods in NLP
Empirical Evaluation

Progression of NL learning tasks


1999 ACL  2001 NAACL

- some ML
- no ML
- reasonable empirical evaluation

- other
- generation
- discourse
- parsing
- lexical
- low-level