A freshman-level, rigorous, non-programming, computer-science intro to NLP, IR, & AI

Lillian Lee
Department of Computer Science
Cornell University
http://www.cs.cornell.edu/home/llee

[This presentation was given at the 2002 ACL Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics, and hence was directed towards an NLP, rather than a computer science, audience.]

Computation, Information, and Intelligence

A new computer-science course on CL/NLP, IR, and AI (henceforth “NLP”).

Three main design decisions:

(1) For entering college freshmen.

Usually junior+ courses (at least at Cornell).

(2) A rigorous, technical focus, including recent research results.

Not “Philosophy of AI” or “The Information Society”.

(3) No programming.

Neither required nor taught.

Motivations and Challenges

(1) For entering college freshmen.

+: Increased early awareness of NLP improves the pipeline.

–: Cannot impose/assume prereqs beyond calculus.

(2) A rigorous, technical focus, including recent research results.

+: Students learn to “do” NLP, not just survey problems and results.

–: Potentially difficult to retain consistency with (1).

(3) No programming.

+: [A CS outreach perspective] A non-programming course may attract a different audience than “CS101” (which already teaches programming)

–: Potentially difficult to retain consistency with (2). [*]

"If they don't know anything, how can you teach them anything?"

Why should such a course succeed?

(1) Students have more relevant background than it may at first seem:

- Today's students are quite familiar with web search (and natural language!).
- Students' *intuitions* about probability often suffice.
- Geometric arguments using just high-school trigonometry abound in machine learning and information retrieval.

(2) Hand simulation on small, well-chosen examples can be an effective replacement for computer-lab exercises.

"If you want to truly understand something, try to change it." – Kurt Lewis

The course format was fairly traditional.

Course material was introduced almost entirely in lecture (no obvious textbook, research papers not suitable; lecture notes available this fall).

Homework involved challenging pencil-and-paper problem sets.

Problems typically investigated implications of lecture material rather than simply testing recall, e.g., students explored the consequences of changing definitions, assumptions, or settings.

Exams were similar to the homework, but emphasized the basic concepts.

Syllabus Outline

"Knowledge without appropriate procedures for its use is [mute], and procedure without suitable knowledge is blind."
– Herb Simon, 1977

Computation: [15 lecs] Search; game-playing; perceptron and nearest-neighbor learning; the halting problem.

Used later: graphs, inner products, Turing machines

Information:

- Document retrieval [3 lecs]
- The World Wide Web [4 lecs]
- Language structure [7 lecs]
- Statistical NLP [6 lecs]

Intelligence:

- The Turing Test [2 lecs]

Document Retrieval [3 lecs]

IR was treated as a subfield of NLP using reduced models of language.

- Tighter integration of the syllabus
- Search engines a highly visible “NLP app”.

Topics: Boolean query retrieval, indexing structures (arrays, B-trees, binary search), the vector space model, term weighting.

Notes: inner products and related geometric notions were introduced in the previous perceptron unit.

The Web [4 lecs]

The Web is a familiar and very large corpus where old and new IR techniques apply.

- What does the Web look like? The "bow-tie" and degree power ("Zipf") laws. [Broder et al 00]
- How can we use explicit link info? Link counting, Google, and the hubs and authorities iterative algorithm [Kleinberg 1998]
- How can we *mathematically model* the Web? The rich-get-richer [Barabasi et al 99] and the copying model [Kumar et al 00]

Notes: recent research is extremely accessible. E.g.,

– The HA algorithm is conceptually very intuitive (skip convergence proof)

– The rich-get-richer result needs only "intuitive probability" and derivatives.

Language Structure [7 lecs]

Bag-of-words $\longrightarrow$ language structure (constituents, heads)

$\longrightarrow$ CFGs (another mathematical model of observed phenomena)

$\longrightarrow$ Pushdown automata (introduces stacks; cf. TMs)

$\longrightarrow$ Grosz/Sidner [1986] discourse theory (super-sentential, AI-complete)

Statistical NLP [6 lecs]

Explorations of sub-sentential, distributional language structure.

Word counts, Zipf's law, and Miller's [1957] monkeys. Same type of argument as the rich-get-richer hyperlink power law derivation

IBM-style statistical MT. Alignments **:** translations **::** hubs **:** authorities.

Japanese segmentation [Ando/Lee 2000]: more multilingual considerations

The Federalist Papers [Mosteller/Wallace 1984]: historical applications

Infant statistical segmentation learning [Saffran et al 1996]: cf. Ando/Lee

Notes: The statistical paradigm was introduced in the unit on learning. Kevin Knight's [1999] tutorial was very helpful.

Statistical NLP [6 lecs]

Explorations of sub-sentential, distributional language structure.

Word counts, Zipf's law, and Miller's [1957] monkeys. Same type of argument as the rich-get-richer hyperlink power law derivation

IBM-style statistical MT. Alignments **:** translations **::** hubs **:** authorities.

...plus other topics ...

Notes: The statistical paradigm was introduced in the unit on learning. Kevin Knight's [1999] tutorial was very helpful.

Statistical NLP (cont)

Japanese segmentation [Ando/Lee 2000]: more multilingual considerations

The Federalist Papers [Mosteller/Wallace 1984]: historical applications

Infant statistical segmentation learning [Saffran et al 1996]: cf. Ando/Lee

The Turing Test [2 lecs]

The Turing Test, the Chinese Room, and the first “Restricted Turing Test” [Shieber 94].

- NLP as AI-complete
- Careful evaluation (philosophically and empirically) is key

Notes:

– Discussing the Turing Test and the Chinese Room *after* seeing what NLP “really is” is much more productive.

– Having already studied Turing machines, proved the Halting Problem undecidable, learned about learning, etc., makes reading Turing’s paper a terrific conclusion.

Results of Pilot Offering

Biggest problem: lack of accessible reference material was frustrating for students. I will be distributing detailed lecture notes this coming fall.

Positives:

- Two Outstanding Teaching Assistant awards, a grant from the GE Fund, and the James and Mary Tien Excellence in Teaching Award.
  - ▷ Student reactions: roughly, “very tough but very interesting.”
- Three research opportunity requests from a class of 23 freshmen (only 1/3 initially considering CS, almost all without prior programming experience.) Currently looks quite promising.

Promising results for potential positive impact on the NLP student pipeline.