Home

CS 4650 and 7650

- **Course**: Natural Language Understanding
- **Instructor**: Jacob Eisenstein
- **Semester**: Fall 2013
- **Time**: Tuesdays and thursdays, 3:00-4:30pm

This course gives an overview of modern statistical techniques for analyzing natural language. The rough organization is to move from shallow bag-of-words models to richer structural representations of how words interact to create meaning. At each level, we will discuss the salient linguistic phemonena and most successful computational models. Along the way we will cover machine learning techniques which are especially relevant to natural language processing.

Learning goals:

- Acquire the fundamental linguistic concepts that are relevant to language technology. This goal will be assessed in the short homework assignments, midterm, and class participation.
- Analyze and understand state-of-the-art algorithms and statistical techniques for reasoning about linguistic data. This goal will be assessed in the midterm, the assigned projects, and class participation.
- Implement state-of-the-art algorithms and statistical techniques for reasoning about linguistic data. This goal will be assessed in the assigned and independent projects.
- Adapt and apply state-of-the-art language technology to new problems and settings. This goal will be assessed in the independent project.
- (7650 only) Read and understand current research on natural language processing. This goal will be assessed in assigned projects and classroom participation.

The assignments, readings, and schedule are subject to change, but I will try to give as much advance notice as possible.

- Schedule
- Grading
- Policies
- Email list

Last edited by jacobeisenstein, 3 days ago

Schedule

- August 20: Welcome: History of NLP and modern applications
  - Reading: Chapter 1 of Linguistic Fundamentals for NLP. You should be able to access this PDF for free from a Georgia Tech computer.
- August 22: Supervised learning: bag-of-words models and naive bayes.
  - Homework 1 due
  - Project 1 out
  - Reading: Chapters 0-0.3, 1-1.2 of LXMLS lab guide
  - Optional reading: Survey on word sense disambiguation
- August 27: Discriminative classifiers: perceptron, logistic regression, regularization, stochastic gradient descent.
  - Reading: Chapters 1.3-1.4 of LXMLS guide
  - Reading: Parts 4-7 of log-linear models
  - Optional reading: Passive-aggressive learning
  - Optional reading: Exponentiated gradient training
- August 29: Unsupervised learning: clustering, expectation-maximization, word sense clustering.
  - Homework 2 due
  - Reading: Nigam et al
  - Optional reading: Expectation maximization
  - Optional readings: Tutorial on EM, Word sense clustering
- September 3: Language models: n-grams, smoothing, speech recognition
  - Project 1 due
  - Reading: Language modeling chapter by Michael Collins
  - Optional reading: An empirical study of smoothing techniques for language models, especially sections 2.7 and 3 on Kneser-Ney smoothing.
  - Optional reading: A hierarchical Bayesian language model based on Pitman-Yor processes. Best for students with strong machine learning background.
- September 5: Finite state automata, morphology, semirings
  - Reading: Knight and May
- September 10: Finite state transduction, edit distance, finite state composition
  - Reading: Chapter 2 of Linguistic Fundamentals for NLP.
  - Reading: OpenFST slides
  - Optional reading: Mohri and Pereira,
  - Homework 3 due
- September 12: Sequence labeling 1: part-of-speech tags, hidden Markov models, Viterbi, B-I-O encoding
  - Project 2 out
  - Reading: Chapter 3 of LXMLS
  - Optional reading: Tagging problems and hidden Markov models
- September 17: Sequence labeling 2: discriminative structure prediction, conditional random fields
  - Homework 4 due
  - Reading: Conditional random fields
  - Optional reading: CRF tutorial
  - Optional reading: Discriminative training of HMMs

- September 19: Sequence labeling 3: the forward-backward algorithm and unsupervised POS induction
  - Reading: Forward-backward
  - Optional reading: Two decades of unsupervised POS tagging: how far have we come?
- September 24: Syntax and CFG parsing
  - Project 2 due
  - Reading: Probabilistic context-free grammars
- September 26: Lexicalized parsing
  - Homework 5 due
  - Reading: Lexicalized PCFGs
  - Optional reading: Accurate unlexicalized parsing
- October 1: Dependency parsing
  - Reading: Characterizing the errors of data-driven dependency parsing models
  - Optional reading: Eisner algorithm worksheet
  - Optional reading: Short textbook on dependency parsing, PDF should be free from a GT computer.
- October 3: Grammar induction and alternative syntactic formalisms
  - Homework 6 due
  - Reading: The inside-outside algorithm
  - Reading: Intro to CCG
  - Optional reading: Corpus-based induction of linguistic structure
  - Optional reading: Much more about CCG
  - Optional reading: Joshi on LTAG
  - Optional reading: Probabilistic disambiguation models for wide-coverage HPSG
- October 8: Midterm
  - Project 3 out
- October 10: Midterm recap. Semi-supervised learning and domain adaptation.
  - Reading: Jerry Zhu's survey
  - Optional reading: Way more about semi-supervised learning
- October 11: Drop deadline
- October 15: Fall recess, no class
- October 17: Compositional semantics
  - Project 3 due
  - Reading: Manning: Intro to Formal Computational Semantics
  - Optional reading: Learning to map sentences to logical form;
- October 22: Shallow semantics
  - Video: Pereira: Low-pass semantics
- October 24: Distributional semantics
  - Homework 7 due
  - Project 4 out
  - Reading: Vector-space models, sections 1, 2, 4-4.4, 6
  - Optional reading: Semantic compositionality through recursive matrix-vector spaces
  - Optional reading: Vector-based models of semantic composition
- October 29: Anaphora resolution
  - Reading: An algorithm for pronominal anaphora resolution
- October 31: Coreference resolution
  - Homework 8 due
  - Reading: Multi-pass sieve
  - Optional reading: Large-scale multi-document coreference
- November 5: Discourse structure
  - Project 4 due
  - Reading: Discourse structure and language technology
  - Optional: Modeling local coherence; Sentence-level discourse parsing
- November 7: Dialogue structure
  - Homework 9 due

- Reading: TBA
- November 12: Project proposal presentations
- November 14: Information extraction
  - Homework 10 due
  - Reading: Grishman, sections 1 and 4-6
- November 19: Phrase-based machine translation
  - Homework 11 due
  - Reading: IBM models 1 and 2
  - Optional reading: Statistical machine translation
- November 21: Syntactic machine translation
  - Reading: Intro to Synchronous Grammars
- November 26: Multilingual learning
  - Reading: Multisource transfer of delexicalized dependency parsers
  - Optional reading: Cross-lingual word clusters; Climbing the tower of Babel
- November 28: Thanksgiving, no class
- December 3: Project presentations
- December 5: Project presentations
  - Homework 12 due at the end of class

Last edited by jacobeisenstein, 7 days ago

Grading

The graded material for the course consists of:

- 12 short homework assignments. Most of these involve performing linguistic annotation on some text of your choice. Each assignment should take less than one hour. You may skip two, but you must do homeworks #10 and #12, which involve giving feedback on the final project. Each homework is worth 2 points (20 total). This grade includes attendance on the due date, because we will discuss them in class.
- 4 assigned projects. These involve building and using NLP techniques which are at or near the state-of-the-art. They must be done individually. Each project is worth 10 points (40 total).
- 1 independent project. This may be done in a group of up to three. It is worth 20 points, including points for the proposal, presentation, and report.
- 1 in-class midterm exam, worth 20 points. Barring a medical emergency, you must take the exam on the day indicated in the schedule.

Students enrolled in the graduate number 7650 will have an additional, research-oriented component to each project assignment.

Homeworks and projects are due at the beginning of class; students should also bring a paper copy of homework and projects to class on the due date. Late homeworks will not be accepted; projects will be accepted up to three days late, at a penalty of 20% per day. This means that a project turned in at the end of class on the due date can receive a maximum score of 8/10 points towards your final grade. This late policy is intended to ensure fair evaluation for everybody.

One of the goals of the assigned work is to assess your individual progress in meeting the learning objectives of the course. You may discuss the homework and projects with other students, but your work must be your own -- particularly all coding and writing. Using external software resources is acceptable unless the assignment directs you not to, but you must clearly indicate which resources you have used. Using other people's text or figures without attribution is plagiarism, and is never acceptable.

Suspected cases of academic misconduct will be referred to the Honor Advisory Council. For any questions involving these or any other Academic Honor Code issues, please consult me, my teaching assistants, or http://www.honor.gatech.edu

Last edited by jacobeisenstein, 3 days ago

Policies

Office hours

Professor Eisenstein's office hours will be 11-12 on wednesday in TSRB 228A. You can TSRB by showing your GTID at the desk. The office is to the left of the stairs on the second floor.

Prerequisites

This course assumes

- Good coding ability, corresponding to at least a third or fourth-year undergraduate CS major. Assignments will be in Python.
- Background in basic probability, linear algebra, and calculus.
- Understanding of automata and formal language theory: finite-state and context-free languages, NP-completeness, etc.
- Experience with dynamic programming algorithms.
- Familiarity with machine learning is *helpful but not assumed*. Of particular relevance are linear classifiers: perceptron, naive Bayes, and logistic regression.

People sometimes want to take the course without having all of these prerequisites. Frequent cases are:

- Junior CS students with strong programming skills but limited theoretical and mathematical background,
- Non-CS students with strong mathematical background but limited programming experience.

Students in the first group suffer in the exam and don't understand the lectures, and students in the second group suffer in the project assignments. My advice is to get the background material first, and then take this course.

Last edited by jacobeisenstein, 3 days ago