Computational Morphology

Lane Schwartz

University of Illinois at Urbana-Champaign

Week 1 of 16

Course web site

`http://computational.linguistics.illinois.edu/lingNNN/SSSYYYY`

- Replace NNN with the course number for this course
- Replace SSS with the current semester (fall or spring)
- Replace YYYY with the current four-digit year

DRES

If a student has a disability or condition that requires special consideration, the student is expected to present the requisite letter from the University Division of Disability Resources and Educational Services (DRES) no later than the beginning of the second day of class.

Github

- Github username

Piazza

- Course Piazza site linked off course web page
- Verify that you are enrolled in the course Piazza site
- Ask questions via Piazza
- Do not ask questions via email

i>clicker

- We will use i>clicker in this class.
- You must have a physical i>clicker.
- You must bring your i>clicker to every class session.
- You must always have working batteries.
- Your i>clicker must be registered.

Academic Integrity

This course follows the University of Illinois Student Code regarding Academic Integrity. The College of Liberal Arts and Sciences also has an excellent web page on the topic. You are expected to read these resources prior to the second day of class, and to understand your responsibilities with regard to Academic Integrity. All work submitted for this class must be solely your own. Violations of Academic Integrity include, but are not limited to, copying, cheating, and unapproved collaboration.

Violations will not be tolerated.

Expectations

- Command line proficiency
- Basic competence with git and LaTeX
- Basic linguistic competence
- Self-motivation to learn and read
- Ability to write in academic style

Computational Morphology

- Motivation for computational morphology
- Computational modelling
- Morphology and phonology
- Finite state toolkits
- Building finite-state models of language
- Applications for morphological analyzers
- Alternative approaches

Motivation

- Why computational morphology?

Computational Modelling

- Set theory
- Relations and functions
- Theory of computation
- Chomsky hierarchy
- Regular languages
- Finite state machines
- Regular expressions

Morphology and phonology

- Study of the shapes and sounds of words

Finite state toolkits

- xfst
- foma

Building finite-state models of language

- Planning
- Lexicography
- Rule building and composition
- Testing and debugging

Alternative approaches

- Two-level phonology
- Supervised learning
- Unsupervised learning

Applications

-
-
-
-
-
-
-
-
-

Applications

- Syntactic parsing
- Text-to-speech
- Spell checking
- Lemmatization
- Speech recognition
- CALL tools / conjugators
- Electronic dictionaries
- Machine translation
- Language documentation

Credits

- Portions of these slides were derived from material graciously made available by Mans Hulden from his Fall 2015 course LING 7800 *Computational Phonology and Morphology* at the University of Colorado Boulder.