Introduction to Methods in Computational Linguistics II

(pre-requisite: Introduction to Methods in Computational Linguistics I)

Prof. Kyle Gorman

SYNOPSIS

This course is the second of a two-semester series introducing computational linguistics and software development. The intended audience are students interested in speech and language processing technologies, though the materials will be beneficial to all language researchers.

OBJECTIVES

Using the Python programming language, students will learn core algorithms used to build speech and language technologies, and best practices for evaluation and basic statistical analysis.

MATERIALS

There is no textbook, but some readings will be assigned. Students are strongly encouraged to bring a laptop computer to the practicum. In some cases, the practicum may be held in the Computational Linguistics Laboratory (7400.13) currently under construction.

ASSIGNMENTS

Assignments will take the form of a small software development project accompanied by a write-up describing the general approach taken and any challenges encountered. Students will often be able to verify the technical correctness of their code by running provided tests. Students will also be graded on the readability of their code, the quality of documentation and the write-up. We will use GitHub Classroom for assignment turn-in.

The final assignment will be an open-ended project which will either extend earlier projects, or build and evaluate a speech and language technology system. Students are encouraged to conceive of projects relevant to their research interests. Students should discuss project plans with the instructor during office hours to confirm that it is both *feasible* and of *appropriate scope*.

GRADING

80% of students' grades will be derived from the assignments; the remaining 20% will be reserved for participation and attendance. Assignments must be submitted on time or will receive a 0 grade (barring a documented emergency).

ACCOMODATIONS

The instructor will attempt to provide all reasonable accomodations to students upon request. If you believe you are covered under the Americans With Disabilities Act, please direct accomodations requests to Matthew G. Schoengood, Vice President for Student Affairs.

ATTENDANCE

Students are extended to attend all lectures and practica. If you are absent for any reason, please contact the instructor in advance with a brief explanation. The instructor reserves the right to tie grades to attendance records. The instructor and teaching assistant are not responsible for reviewing materials missed to absence. Assignments must be submitted on time or will receive a 0 grade (barring a documented emergency).

INTEGRITY

In line with the Student Handbook policies on plagiarism, students are expected to complete their own work. However, a student is permitted to collaborate with another student during the coding phase of an assignment so long as they: do not share lines of code with each other, *mutually* disclose their collaboration in their write-ups, and do not collaborate *at all* on their write-ups.

The instructor reserves the right to refer violations to the Academic Integrity Officer.

RESPECT

For the sake of the privacy, students are not permitted to record lectures. Students are expected to be considerate of your peers and to treat them with respect during discussions.

SCHEDULE

(Please note that this is subject to change.)

Syllabus & motivations
Git & GitHub
Program design & structure
Software testing
Probability theory
Formal language theory
Finite-state transducers
Language modeling
Finite-state grammars
Generative classification
Hidden Markov models
Discriminative classification
Evaluation

REFERENCES

Bird, S., Klein, E. and Loper, E. n.d. *Natural Language Processing with Python*. URL: https://www.nltk.org/book/.

Breiman, L. 2001. Statistical modeling: The two cultures. *Statistical Science* 16(3): 199-231.

Chacon, S., and Straub, B. 2014. *Pro Git*. 2nd edition. Apress. URL: https://git-scm.com/book/en/v2.

Collins, M. 2002. Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)* , pages 1-8.

Gorman, K. and Sproat, R. In press. *Finite-State Text Processing*. Morgan & Claypool.

Resnik, P. and Lin, J. 2010. Evaluation of NLP systems. In Clark, A., Fox, C., and Lappin, S. (ed)., *The Handbook of ComputationaL linguistics and Natural Language Processing*, pages 271-295. Wiley-Blackwell.

Freund, Y., and Schapire, R. E. 1999. Large margin classification using the perceptron algorithm. *Machine Learning* 37(3): 277-296.

Jelinek, F. 1997. *Statistical Methods for Speech Recognition*. MIT Press.

Jurafsky, D., and Martin, J. H. 2009. *Speech and Language Processing*. 2nd edition. Pearson.

Manning, C. D., and Schütze, H. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

Ng, A. Y. and Jordan, M. I., 2002. On discriminative vs. generative classifiers: A comparison of logistic regression and naive Bayes. In *NeurIPS*, pages 841-848.

Partee, B. H., ter Meulen, A., and Wall, R. E. 1993. *Mathematical Methods in Linguistics*. 2nd edition. Kluwer Academic Publishers.

Roark, B. and Sproat, R. 2007. *Computational Approaches to Morphology and Syntax*. Oxford UniversityPress.