.

# LING 248 Topics in Computational Methods and Models

Syllabus (as of 01.05.16)

## 1 Administrative Information

INSTRUCTOR:
Pranav Anand
Location: Stevenson Rm. 260
E-mail: panand@ucsc.edu
Phone: 831.459.3352
Office hours: Tu 2-3

LOGISTICS:
Class: Tu 10-11.45, F 16-17.45 [Th 14-15.45]
Website: `github.com/panand/Ling248_2016`
Prerequisites: Syntax A and Semantics A, their equivalent, or instructor's permission.

THE PLAN

This course is designed to introduce students to material at the boundary of computational linguistics and theoretical linguistics. Although topics vary, in this instantiation, we will focus on trends in computational work in semantics. Based on our collective wisdom, this what we will be covering:

1. anaphora resolution (discourse anaphora, ellipsis)
2. semantic/propositional relations (entailment, similarity)
3. "new" methods (distributional semantics, deep learning)

However, before we get to that, I would like to spend some time providing some foundational material in computational linguistics. To that end, we will devote the first few weeks to the following units:

1. probabilistic foundations
2. information theory
3. language models
4. word sense disambiguation
5. markov models
6. probabilistic parsing
7. dependency parsing

## 2 Requirements

The course has six formal requirements: assigned readings, presentation of class readings, documenting class knowledge, homework, a course project, and a final paper. All of these requirements must be met for satisfactory credit.

## 2.1 Assigned Readings

We will likely read 1-2 papers per week, or their equivalent in books. This reading is likely to be mathematically technical in a way that will require close attention. Please be prepared for this pace.

## 2.2 Presentation of Class Readings

You will have to present a part of **2 readings** from the course list. We will decide on this schedule at the beginning of the course. Note that the presentations are designed to be short (30 minutes) and focus on some particular aspect of a reading (exploring or explaining the techniques used, discussing some of the background reading that the paper assumes, etc.) and I will direct you to what salient aspect to focus on.

## 2.3 Documenting Class Knowledge

To keep track of what we have learned, each week I will designate a scribe, whose job it is to summarize in a few paragraphs (maybe a page?) what we learned over that week about the central issues of the course. The idea is for this to be a dynamic process, and my hope is that we can use later readings to reflect on earlier readings.

## 2.4 Homework

Because of the technical nature of this course material, it is imperative that you try out small-scale exercises on the techniques covered. I will assign very short programming assignments approximately every week to explore the data, techniques, and problems currently being discussed. These will be designed to take no more than 4 hours to complete.

## 2.5 Course Project

You will have to complete a data-driven project for this course by **March 11th**. The project will ideally grow out of one of the homework exercises you undertook, and should build on that to arrive at a small codebase and a project report of 5-10 pages, single spaced, 12 point font. Ideally, this should take you a week's worth of work. I will list potential project ideas alongside each homework, and we will keep a running list of ideas on the course website.

## 2.6 Research Paper

You are required to write a final paper, due on **March 19th**. The paper should be 10-20 pages, single spaced, 12 point font. The paper must connect with the ideas of the course. If you are concerned about the suitability of your topic, come talk to me well in advance. Ideally, this paper should grow out of your project, extending it in useful and interesting ways.

A one-page **proposal** for the paper, consisting of a description of the problem you are working on and how you plan on explaining it, will be due on **March 7th**. We will likely discuss the proposals in class that week.