LING 5370/4371: Intro to Computing Natural Language
Fall '19

Professor: Nicholas Sobin; LART 113; 747-7023; njsobin@utep.edu

Office hours: 10-11 a.m. M-Th, and by appointment

Text: Sobin, *Computational Linguistics: A Hands-On Introduction*. (Available on Blackboard)

Other materials: a flash drive, or other appropriate storage medium

Some questions: Here are a couple of interesting questions: First, how is a human language built? That is, when someone 'knows' a language, what exactly does that person know? Second, how would you go about making a machine produce (speak/understand) a human language? In this course, while not being able to give comprehensive answers to these questions, we will deal with at least certain basic aspects of them.

The first question has to do with the nature of human language itself. People commonly use the term 'grammar' to refer only to prescriptive classroom grammar -- the *do*s and *don't*s of writing formal English. For linguists/language analysts, or for anyone who has to deal in the real substance of a human language like English or Spanish, such a characterization is useless. Prescriptive grammar does not begin to reveal the real workings of a human language. A different concept of grammar, one which is much more useful in understanding the nature of human language, can be characterized as follows:

> The ***grammar*** of a language is the subconsciously known system which allows the speaker of a language to produce and understand utterances in that language.

This sort of grammar is also called ***linguistic competence***. It is this sort of grammar that the first question above is asking about--it is an inquiry into the nature of linguistic competence.

The second question logically presupposes an answer to the first. A human language is infinitely large. That is, there are infinitely many sentences in it. Therefore, we can't put its sentences directly into a machine (or into a person's head, for that matter). It must be the grammar (linguistic competence) which produces these sentences that speakers somehow learn and that we would have to 'program' if we are to make a machine process a language anything like a human being does.

Goals: In this course we will (i) investigate some of the basic architectural aspects of the grammatical system of English, in particular, systems of word structure and sentence structure, and (ii) learn how to create computational models of the language which emulate its linguistic architecture. We hope to accomplish a number of specific objectives in this course, some of which are linguistic and others of which are computational. You should acquire:

- key terms/concepts related to morphology, syntax, and Prolog computation;
- knowledge of some classic generative analyses of lexical and syntactic constructions;
- the ability to operate a given syntactic or morphological analysis;

•the ability to operate and map/'tree' a given Prolog analysis;
•some ability to argue for the superiority of one linguistic analysis over another and of one Prolog analysis over another;

We currently plan to meet for the first half of the class in the classroom where we'll discuss what we will be analyzing, and deal with any initial questions. We will hold the second half of the class in a computer lab. We will start with an introduction to the basics of doing this sort of work, after which you should be able to begin tackling the text and accompanying exercises. I will be in the lab to assist you in any way that you need. In the lab, we will try to maintain a relatively free work-shop-like atmosphere. You are welcome to consult with each other as well as with me. You must read the assigned material outside of class in preparation for the classroom discussion and doing the in-lab programming.

Course work: Work for the course will include reading assignments, discussion, and working problems in grammatical/computational analysis, both on and off the computer. **Keep up with the work.** Late assignments will not be accepted without prior arrangement. **Participate.** Get actively involved in our discussions, and ask questions. I also learn things from your questions. **Attend class.** Regular attendance is crucial to this work. More than two absences (6 hrs) may, at the discretion of the professor, result in being dropped from the course, as indicated in the university catalogs (both graduate and undergraduate).

Grade: The major work of the course will be the in-class computer work. I will take up assignments **at the end of the lab sessions**, and these will constitute 50% of the grade. I will work with you as much as I can to get the computer exercises working "well." In addition, there will be a couple of quizzes/exams during the course of the semester worth 25% each. These will ask about such things as terms, general ideas, operating linguistic and computational analyses, and arguments for particular analyses. Students taking the graduate level course will complete an additional programming project (subject to approval by the professor), due on the last class day. This project will count for 20 % of the final grade, with the above components comprising the remaining 80%.

Some final points: Since much of the course will be in the lab, there will be some simple rules of course and lab etiquette that it is important to observe, so that the course can accomplish its objectives.

• Please have cell phones turned off during class and in the lab--no talking or texting. Cell phones are strictly forbidden in exams.

• When in the lab, please do the work of the course. Assignments will be due at the end of the lab sessions.

• Please do not come late.

• Please do not leave before the end of the class unless you have completed the lab

assignment for that period.

Finally,

• I encourage you to discuss the things that we are working on with others in the class, but please do your final versions of assignments individually. **All submitted work must be yours and yours alone**.

Violations of any of these rules may be grounds for a reduced or failing grade, or being dropped from the course, at the discretion of the professor.

This is a tentative sketch of the structure and grading of this course. It may be modified in any aspect as circumstances or other considerations dictate.

Major topics by chapter:

Chapter 1 Intro to programming in Prolog
Predicates and arguments; True/false in a given world; Matching/unification; Logical connectives and truth; Free and bound variables; Creating logic-based programs in Prolog; Glossary of some Prolog terms for later reference.

Chapter 2 Lists in Prolog
List manipulation in Prolog; Membership in a list; Concatenation of lists.

Chapter 3 Analyzing words with rules
General terms; Rule-based morphology; Roots, affixes, and stems ; Inflection vs. derivation; Other aspects of regular affixation.

Chapter 4 Paradigms, irregular morphology
A problem with irregular verbs; Inflectional paradigms; Present tense forms reconsidered; Subtypes and other extra designations in Prolog; Adding present participle; The regular past forms; Irregular past morphology; Regular and irregular past participle forms; The noun paradigm and irregular morphology.

Chapter 5 Various approaches to computing sentence structure
Word-class sentence patterns; Transition networks (finite-state grammar); The phrase structure grammar; Phrase types & tests of constituency.

Chapter 6 Proforms; coordination; sentence embedding
Analysis of pronouns(case, person, number, & gender); Pronouns/ProNP as evidence for NP; Other proforms & other phrases; General dimensions of coordination; Recursive NP coordination & depth-bounding; Recursive sentence embedding.

Chapter 7 Argument structure; possessive expressions
Transitivity; Subcategorization; Argument structure; thematic roles; Recursive possessive expressions

Chapter 8 Case; agreement; ambiguity
Subject-verb agreement and case forms; Adjuncts vs. complements; Structural ambiguity; Bracket trees in Prolog

Chapter 9 Category-neutral syntax; X-bar syntax
Category-neutral coordination; X-bar analysis of NP

Chapter 10 Displacement
Basic wh-movement and subject-auxiliary inversion; Superiority; Referential indices & traces; Passivization; Case and thematic roles in active and passive.

Disability Statement:

If you have a disability and need classroom accommodations, please contact The Center for Accommodations and Support Services (CASS) at 747-5148, or by email to cass@utep.edu, or visit their office located in UTEP Union East, Room 106. For additional information, please visit the CASS website at www.utep.edu/CASS.