Introduction to Computational Linguistics
Ling 581

Jean Mark Gawron
San Diego State University
gawron@mail.sdsu.edu
http://gawron.sdsu.edu/compling

2017 Jan 20

1 Introduction

This course will serve as an introduction to the field of computational linguistics, which includes aspects of speech recognition, natural language processing, information retrieval, and information extraction.

The course begins with an introduction to finite-state automata and some basic natural language applications; this is extended to finite-state transducers with applications in morphology (word structure). Other topics covered: ngram language models, classifiers (Naive Bayes and Logistic Regression), sentiment analysis, part of speech tagging, context-free grammars and context-free parsing (with statistical extensions), and distributuional semantics.

2 Goals

The primary goal of the course is to acquaint students with a basic set of computational techniques that have proved useful in a variety of natural language applications, with a particular emphasis on probabilistic methods. The principles and mathematics behind these techniques often overlap with

those used in other fields in which machine learning has been successfully applied, such as computer vision. However, the problems, in particular the relevant statistical properties, are quite different. Thus, this class should provide a nice complement to other classes you may be taking which use machine learning.

Students should acquire enough facility with the concepts and tools so that they can use them to construct well-specified solutions to simple computational linguistic problems. A well-specified solution is one that a programmer can use to write a program.

3 Practice

The course will use the textbook:

> Jurafsky, Daniel and Martin, James H. 2000. *Speech and Language Processing*. Prentice-Hall. (2nd Edition only!)

There will be exercises for most of the chapters covered.

4 Classroom Practice

Assignments will generally be due on a Tuesday and will be discussed upon return. Model solutions will often be posted.

Most of the readings are from the 2nd Edition of your textbook, but some are available ONLY in the 3rd Edition, which has not yet been published.

5 Pre-requisites

At least two linguistics courses or at least two programming or CS courses. Students with no programming background will find this course challenging.

6 Grading

Grading will be based on exercises, a midterm, and a final.

Exercises	50%
Takehome Midterm	20%
Takehom Final	30%

7 Late Assignments

The general structure of the course is not well-suited to late assignments or missed quizzed. Assignment solutions will be discussed in detail on the day they are turned in, and thus students who turn assignments in late will be at an advantage. Quizzes are designed to test understanding of foundation needed for further work, and without those foundations, progress will be slowed. However, to allow for some flexibility, late assignments will receive partial credit. Here is the lateness policy:

1. Up to one week late: 50% credit for assignment . Late assignments must include all problems for which solutions have not been posted in order to receive any credit at all.

2. More than one week late: not accepted

8 Attendance

Attendance is not a formal part of your grade.

However, be aware that assignments are, and extensive amounts of class time will be devoted to working through exercises like those on the assignments. Similarly, hints on how to solve problems on the assignments and the midterms are handed out liberally in class. These hints will not be posted on the web pages.

9 Group Work

Group work is encouraged on the assignments. The midterm and final project should be completed without any help. To be clear on this, collaboration or group work on the midterms and finals will be considered cheating.

When turning in collaborative assignments, your collaborators should be identified on your paper. The code you write on your group assignments should be your own.

10 Learning Outcomes

1. Students will able to identify and use three different incarnations of Finite-State Natural Language processing methods, Regular Expressions, Finite-State Automata (as language recognizers and generators), and Finite-State Transducers (as morphological analyzers).

2. Students will be able apply basic laws of probability to derive three different kinds of applied probabilistic language models, ngram models, HMM taggers, and Naive Bayes Classifiers (applied to sentiment analysis).

3. Students will learn the main probabilistic ideas embodied in Naive Bayes and Maximum Entropy Classifiers.

4. Students will be able to identify the major components in a CKY context-free parser, to sketch the proof that it is an n^3 algorithm, and to understand how an exponential number of parses can be stored in n^2 space.

5. They will also learn how to convert an all paths CKY parser into a statistical best-parse parser.

6. Students will be able to identify dynamic programming aspects of three major algorithms, Minimal Edit Distance, the Viterbi Best Path Algorithm (for HMMs), and the CKY context-free parsing algorithm.

7. Students will be able to explain the geometric justification for a variety of semantic similarity models for words.

11 Office Hours

Th	9:30-10:30
Tu,Th	11:00-12:00
Tu	3:30-4:30
	by appointment

12 Mailing address

Dept:	Department of Linguistics and Oriental Languages
Uni:	San Diego State University
Address:	5500 Campanile Drive
City:	San Diego, CA 92182-7727
Telephone:	(619) 594-0252
Office:	SHW 238
Email:	gawron@mail.sdsu.edu

13 Weekly Syllabus

`http://gawron.sdsu.edu/compling/course_core/course_outline.html`