Syllabus for
CS 287: Statistical Natural Language Processing

1 Overview

CS 287r is a graduate introduction to statistical natural language processing, i.e. the analysis and transformation of written language by computational methods. Natural language processing (NLP) aims to create general representations of text that can aid prediction, extraction, and semantic reasoning over language. Recent consumer developments in NLP include automatic language translation, hand-held personal digital assistants, and the extraction of structured knowledge bases from the web.

Modern statistical NLP is highly intertwined with the field of machine learning (ML). NLP provides a rich collection of challenging and large-scale applications , whereas ML provides a formal vocabulary and set of techniques for statistical modeling. Over the last two decades developments in both areas have led to major measurable progress towards computational understanding of language.

The interplay between ML and NLP will be the central focus of this class. Much of the course will concentrate on developing applied mastery of the central models and algorithms of machine learning, such as multinomial models, multi-class logistic regression, hidden Markov models (HMM), and, particularly this year, a focus on deep learning/neural network models such as log-bilinear models (LBL), convolutional networks (CNN), and recurrent neural networks (RNN). We will explore the various benefits and downsides of these models with both mathematical analysis and hands-on experimental work.

Despite the general effectiveness of these methods, throughout the class we will often see that the naive application of ML to NLP often leads to significant computational and modeling challenges. For some of these challenges we will see clean eloquent solutions; for others we will see only open questions and opportunities for new developments.

In particular, three core properties of text will guide the overarching questions of our study: (1) Text is a discrete system. What higher-level representations of text can be used by statistical models? How can these representations be improved by data-driven approaches? (2) Text is a structured system. How can we go beyond system of regression and classification to predict complete structured outputs such as translations or syntax? (3) Text is a symbolic system. How do we account for phenomenon like reference and dependency relationships within a sentence or document? When is it necessary to model these directly? By the end of the class students will have seen several different approaches for dealing with these challenges, and will be able to develop their own methods for overcoming them.

Objectives Students completing this course will have a the background to read, implement, and extend state-of-the-art research in natural language processing. They should be able to:

- develop formal models to express natural language phenomenon
- utilize mathematical language to describe algorithms for language processing
- implement and debug large NLP systems in a clean and structured manner
- design and analyze the computational performance of the algorithms presented in the class
- describe the results of statistical systems in a logical and empirical way, both in writing and orally
- critically read papers from NLP and ML conferences

Finally, the main assignment for the class will be a final project due in May. We expect the final project to be a significant research project aspiring to conference publication level. We note that there are often conference deadlines for EMNLP and NIPS (ML) in early June.

2 Preliminaries

Prerequisites CS 181, CS 281, or Stat 110, as well as significant programming experience. No previous exposure to NLP is assumed. Talk to the instructor if you're concerned about your preparation. Programming assignments will use the programming language Lua and the Torch framework. We do **not** expect any familiarity with Torch/Lua, but think of it like NumPy or MatLab. We expect written assignments to be submitted in LaTeX.

Textbook The course will not have a textbook. We will draw heavily from a set of free course notes available online. For background in machine learning, we recommend **Machine Learning: A Probabilistic Approach** which is available in the COOP, and is a generally worthwhile reference. We will also read several research papers during the term. Finally, the course staff is always happy to recommend additional readings or other sources of information if you would like to explore a topic from the course in more depth.

Laptop Policy For the sake of cutting down on distraction and maintaining an academic atmosphere, phones and laptops will in general not be permitted during lecture. As we understand that some students prefer laptops for note-taking, we ask that you contact the course staff at the beginning of the semester if you require your laptop during class. We will also ask that students using laptops sit in a designated section so as not to distract other colleagues.

Support resources We will be using the Piazza for questions. Unless your question would reveal confidential information or give away answers to homework questions, please post there. We also encourage you to answer each others questions.

Office hours The staff office hours will be posted on the website. You are welcome to come with specific questions about the material, to discuss final project ideas, or just to chat about things you find interesting and want to explore further.

Email Staff emails are posted on the website. To avoid duplication of questions and keep the email load manageable, please use the forums if your question may be of interest to other students, and only use email for personal questions.

3 Provisional Schedule

This weekly schedule is provisional. It may be adjusted based on the observed pace of the course:

Date	Topic	Lecture	Assignment
Jan. 26	*Natural Language Processing*	Tasks	
Jan. 28	Text Classification	Naive Bayes	HW 1
Feb. 2		(Multinomial) Logistic Regression	
Feb. 4		Optimization Methods	
Feb. 9	Neural Networks	Fundamentals	HW 2
Feb. 11		Neural Text Classification	
Feb. 16		Convolutional Neural Networks	
Feb. 18	Language Modeling	Multinomial Models	HW 3
Feb. 23		Neural Language Modeling	
Feb. 26		Embeddings	
Mar. 1	Midterm		
Mar. 3	Application: Coreference		
Mar. 8	Recurrent Neural Networks	Fundamentals	HW 4
Mar. 10		LSTMs and Language Modeling	
Mar. 22		Approximate Search	
Mar. 24		LSTMs and Machine Translation	
Mar. 29	Structured Prediction	Sequence Models	
Mar. 31		Filtering, Smoothing, Viterbi	HW 5
Apr. 5		Conditional Random Fields 1	
Apr. 7		Conditional Random Fields 2	
Apr. 12		Dependency Parsing	
Apr. 14	Topics	Lagrangian Relaxation	
Apr. 19		Attention-Models	
Apr. 21		Guest Lecture	
Apr. 26	Final Project Presentations		

4 Course Requirements

The course has several components:

- Six assignments; each will have a computational part and written part (40%)
- An in-class exam (10%)

- A final project (40%)
- Paper discussion and scribing (10%)

Final grades take into account each component. You must achieve a passing grade in all components to pass this course. To receive an A you must have high performance in all categories.

Assignments The 5 assignments (HW 1 - HW 5) will be published on the course webpage. Most assignments have two components: computational and written. The computational part can be done in pairs or individually and will require programming and experimentation. The written part will be submitted individually and will include analysis of the results. Computational assignments will ask you to develop implementations of algorithms for multi-class classification, neural network classification, language modeling, and generation with an LSTM, and training a conditional random field. You be expected to apply them to different real-world problems, and to analyze the performance in a write-up.

Late days Each student is allotted **five** late days which may be applied to any of the assignments. A late day extends the due date by 24 hours. No more than two late days may be used on any one assignment. In cases of medical or other emergencies which interfere with your work, please contact the instructor.

Grading Assignments will be due at **5pm** on the day scheduled. If you have used up your 5 late days, you will be penalized 25% per day, up to two days max, with no credit after two days. We will only give extensions for emergencies, and you will need a note from either a doctor or your Resident Dean. Computational components will be graded based on correctness, performance and documentation. Written components will be graded based on correctness, depth of analysis, and clarity.

Participation, Discussion, and Scribing We will have several paper discussions sessions. You are expected to read the paper before class and engage in discussion in the class itself. We will offer two sources of additional credit in the class for participation. First for students who go out of their way to provide support in the Piazza forums, and second for students who find bugs or errata in the course lecture notes or homeworks. For the second, please email the professor with the subject `CS287 Errata` in the subject line. Finally each student will also be asked to scribe one class in the semester. The scribe will be required to provide a complete TeX'd version of the notes for that lecture.

In-class Exams In addition to homework assignments, there is one in-class exam (closed book, no notes), covering the first half of the course material. See the schedule for dates and topics covered.

Final Project During the course students will design and carry out a final project, working in pairs. The final project is of your choosing, but we expect it to aspire to the level of a conference publication. We will provide a list of potential topics and an opportunity to get feedback before

starting. The final presentation and paper (by the group) are due at the end of reading period, and attendance at the final presentation sessions is mandatory.

The project grade is based three aspects:

1. project concepts and results
2. presentation quality
3. final paper quality

Collaboration Policy Each assignment will include a computational component and a written component.

The computational component of assignments 1-5 can done and submitted in pairs. In pairs implies designing and writing the code together and submitting a single assignment and receiving the same grade. Note that we will treat pairs/non-pairs the same from a grading perspective. We expect you and your partner to design and implement the solutions together. You may also consult with your classmates in other groups as you work on the problem, but you should not talk in terms of pseudocode or real code, and you should not share answers. In addition, you must cite any books, articles, websites, lectures, etc. that have helped you with your work. Similarly, you must list the names of students from other groups with whom your group has collaborated. If you are doing the computational assignment individually, then the same rules apply for collaboration as for the written assignments: talking is ok, sharing code is not.

The written component of all assignments must be done individually, and each person must submit her/his own written assignment. You are encouraged to consult with your classmates as you work on the problems for the written assignments. However, you should not share answers. After discussions with your peers, make sure that you can work through the problem yourself and ensure that any answers you submit for evaluation are the result of your own efforts. In addition, you must cite any books, articles, websites, lectures, etc. that have helped you with your work. Similarly, you must list the names of students with who you have collaborated. Note that understanding the concepts in the written assignments is important both for the computational components and the exams. Final projects must be done in pairs. You are encouraged to discuss your project ideas with your peers.

For any questions not covered in this document, email the course staff for clarification.