Syllabus for
CS 287: Machine Learning for NLP

1 Overview

CS 287r is a graduate seminar on machine learning for 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 topic of focus this semester will be on *text generation and latent variable models*. We will study recent advances in text generation with neural networks and the potential of machine learning methods, in particular deep denerative models, to untangle the structural factors of generating natural text:

The first half 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, and, particularly this year, a focus on deep learning/neural network models such as log-bilinear models, 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.

The second half of the course will focus on research projects particularly in text generation and modeling. 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 This is a challenging and fast-paced course. CS 181 and Stat 110 are required, CS 281 is recommened, significant programming experience in Python or Matlab is necessary. 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 Python and the PyTorch framework. We expect written assignments to be submitted in LaTeX.

Textbook The course will not have a textbook. We will have several papers associated with each class, that are required reading. We will also draw heavily from a set of free course notes available online. For background in machine learning, we recommend **Machine Learning: A Probabilistic Approach**, textbook from CS281, which is available in the COOP, and is a generally worthwhile reference. 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 Laptops are allowed.

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

The schedule is posted on the website. It may change over the course of the semester.

4 Course Requirements

The course has several components:

- Assignments (20%)
- Presentation and Participation (15%)
- Midterm Exam (15%)
- Final Project (50%)

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 4 assignments (HW 1 - HW 4) will be published on the course webpage. Most assignments have two components: computational and written. Both parts should be done in pairs and will require programming and experimentation. Computational assignments will ask you to develop implementations of algorithms for multi-class classification, neural network classification, language modeling, and generation. 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 in class 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 and Discussion We will have several paper discussions sessions. You are expected to read the paper before class and engage in discussion in the class itself. Additionally each group will present papers in class.

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 Collaboration is welcome. Students will complete assignments in teams of up to 3 people.