Syllabus: CS395T Structured Models for NLP

Instructor: Greg Durrett, gdurrett@cs.utexas.edu
Instructor Office Hours: Wednesdays 10am-12pm in GDC 3.420
TA: Ye Zhang (see website for office hours)

1 Course Description

The last 25 years of progress on building AI systems that deal with language data has largely been driven by statistical machine learning. Because language is a fundamentally structured thing, the most successful models for this task are structured: assumptions they make and dependencies they impose all reflect sequential, tree, and graph structure of language. This is true for both "conventional" linear models as well as neural network models.

This class covers a range of models in structured prediction and deep learning with applications to NLP. We discuss model structures that commonly arise in NLP such as sequence models, tree-structured models, more general graphical models, recurrent neural networks, convolutional neural networks, and connections between these. We study the models themselves, examples of problems they are applied to, inference methods, parameter estimation (both supervised and unsupervised approaches), and optimization. Programming assignments involve building scalable machine learning systems for various NLP tasks, with a focus on understanding design decisions surrounding modeling, inference, and learning, and how these interact.

Differences from CS388: This class is intended to complement CS388; CS388 is not required as a prerequisite for this class, nor will those who have taken CS388 have seen everything in this class. In particular, this class has a greater emphasis on the fundamentals of structured machine learning and covers a wider range of deep learning techniques, while CS388 deals more with covering broadly important problems in NLP and studying the underlying linguistic phenomena.

2 Prerequisites

- 391L - Machine Learning (or equivalent)
- 311 or 311H - Discrete Math for Computer Science (or equivalent)
- Familiarity with Python (for programming assignments)
- Additional prior exposure to probability, linear algebra, optimization, linguistics, and NLP useful but not required

3 Lectures

Lectures are 9:30-11:00am Tuesday and Thursday in Garrison Hall 0.132 (GAR). A complete schedule of lectures and assignments, complete with readings, is on the course website:
http://www.cs.utexas.edu/ gdurrett/courses/fa2017-cs395t.shtml

There is no required textbook for this course. Optional readings from book chapters and papers will be posted on the course website.

4 Assignments

The work for this course consists of 3 programming projects (16.6% of your grade each, 50% total) and a final project (50% of your grade).

4.1 Projects

Three programming projects will be assigned in this course. Projects are downloadable from the course website; due dates are listed there as well. Framework code is provided in python, which you are strongly encouraged to use. However, if you wish to use another language and recreate the framework code from scratch, you may.

Each project centers around an NLP task on a standard dataset, with part of the project being an open-ended extension to allow you to explore what's most interesting to you about the data, model, training procedure, etc.

Projects will be submitted on Canvas. Submissions should include your writeup, a gzipped tar or zip file of code, and output on the blind test set.

Project Writeups For each project, you should submit a report of around 2 pages not including references (though you aren't expected to reference many papers). Your report should restate the core problem and what you are doing, describe relevant details of your implementation, present results, describe your extension, and optionally discuss error cases addressed by your extension or describe how the system could be further improved. Your report should be written in the tone and style of an ACL/NIPS conference paper. Any format with reasonably small ($\approx$1" margins) is fine, including the ACL style files (available at `http://acl2017.org/calls/papers/`) or any one- or two-column format with similar density.

Collaboration You are free to discuss the homework assignments with other students and work towards solutions together. However, **all of the code you write must be your own!** Your extension should also be distinct from those of your classmates and your writeup should be your own as well.

Grading Assignments will be graded on a 10-point scale.

- 6 points for minimally complete code
- 1 point for a minimally complete writeup
- 1 point for a minimal extension
- 2 points for a more interesting extension, higher quality writeup, etc.

Grades are assigned based on the weighted average of the three project grades and the final project according to the following scale:

$10-9$	$9-8.5$	$8.5-8$	$8-7.5$	$7.5-7$	$7-6.5$	$6.5-6$	$6-5.5$	$5.5-5$	$5-0$
A	A$-$	B+	B	B$-$	C+	C	C$-$	D	F

Grades on the boundary receive the higher grade (9 is an A). Therefore, a minimally completed project will receive an A$-$, and a well-done project will receive an A.

Slip Days Each student is given 7 slip days to use throughout the term. Any number of these days can be applied to any project to extend the deadline These days can be applied to any of the projects. E.g., you can turn the first project in 2 days late, the second 5 days late, and the third on time.

4.2 Final Project

The final project is an opportunity for open-ended exploration of concepts in the course. This project should constitute novel work beyond directly implementing concepts from lecture and should result in a report that roughly reads like an NLP/ML conference submission in terms of presentation and scope. You may work on the final project either individually or in groups of two.

Proposal You will write a brief proposal (<1 page) explaining your idea. Individual meetings with the instructor may be scheduled for each group, TBD based on class size.

Writeup Your final project report should be 4-8 pages—use your discretion about the length. Groups of two should have reports closer to 8 pages. The scope should be similar to that of an ACL paper: you should present a novel idea, discuss related work, describe your implementation or what you did, give results, and provide discussion or error analysis.

Slip Days Slip days will not be permitted for the final project. The final project is due on Friday, December 15.

Presentation Depending on class size, you may be asked to give lightning talks (<5 minutes) about your project on the last class day.

4.3 Compute Resources

The assignments are largely designed to be doable on personal computers (assuming you write your code efficiently!). However, for extensions and for your project, you may wish to run longer experiments. We encourage you to do so using the department's Condor pool. An overview of Condor can be found here: `http://www.cs.utexas.edu/facilities/documentation/condor` and some documentation can be found here: `https://www.cs.utexas.edu/~ml/faq/mastodon_readme.html`

Be aware that your jobs may be terminated by Condor if they are competing for resources and plan ahead for this if you choose to use Condor.

5 Miscellaneous

Disabilities Students with disabilities may request appropriate academic accommodations from the Division of Diversity and Community Engagement, Services for Students with Disabilities[1] at 471-6259.

[1] On the web at `http://ddce.utexas.edu/disability/`