

Course Missive

Spring 2021

Time and Location: MWF 2:00 - 2:50 PM, Online

TA email: cs1460tas@lists.brown.edu

Course Staff

What	Who
Professor	Eugene Charniak (ec)
Undergraduate TAs	Tucker Berckmann (tberckma) Luca Fonstad (lfonstad) Tian Yun (tyun)

Introduction

Welcome to CS146! Have you ever wondered how a computer program can complete such amazing tasks as:

- translating English texts into exotic foreign tongues,
- identifying the subject, verb, and object of a sentence,
- understanding the semantics of text and converting it to actions, and
- generating new sentences from scratch,

all with little or no human instruction? Do you want to take the first step on the road to becoming Professor Eugene Charniak? If so, this is the course for you!

Due to the rapid development of Computational Linguistics as a field, this course will focus on recent advancements of NLP, including classic papers from the past few years and new papers that are just released.

This course satisfies the Artificial Intelligence/Machine Learning pathway requirement for the Sc.B. or A.B. track in Computer Science.

Prerequisites

Students should have taken CSCI 1470 (Deep Learning), or have an equivalent deep learning background.

While not formal prerequisites, students should also have a basic programming background (equivalent to CS15/16, CS17/18, or CS19).

FAQs

1. Q: What Deep Learning libraries can I use?

A: The recommended library is PyTorch, although you are free to use TensorFlow if you prefer. We have a warm-up lab aimed at helping those who are unfamiliar with PyTorch learn how to use it. Any code that can be run on our specified environment, which includes TensorFlow (2.0) and PyTorch (1.4), will be accepted. We are releasing those environment specifications on the course website.

2. Q: Do I need to know Linguistics? How much Deep Learning experience do I need?

A: No. You don't need Linguistics knowledge to take the course, but you do need to know the basics of Deep Learning for NLP. If you are not familiar with LSTMs or Transformers, we would not recommend that you take the course. If you are still unsure, email the TAs and we will help you.

3. Q: Can we go over a paper that I find interesting?

A: Bring it up in class, or email Prof. Charniak in advance. If he think it's worthy to go over in class, he most likely will.

Grading

Participation	10%
Programming Assignments	60%
Final Project	30%

Assignments

Assignments are short programming projects that involve replicating the experiment results in contemporary research papers. In addition to the code, we ask you to respond to some written to questions about your work. There will also be a final project at the end of the semester.

Unlike programming assignments in many other courses, the nature of the projects in this course will be more exploratory and open-ended. While students are required to follow the experiment in the paper to replicate its results, most of the papers do not list exact implementation details, meaning students will be given freedom in the approach of the implementation. Extra credit may be given if students make a novel addition to improve the results, or implement features beyond the part of the paper that was assigned.

In some cases, achieving results comparable to the paper may be a difficult task: given that implementation details are often ambiguous, it is expected that students will try out different implementations that may not end up working out. Each assignment will thus have a written component, where students will document what they have tried and haven't

tried, the experiment results of the various implementations they have attempted, hypotheses on why the implementations work or don't work, and what other approaches they envision trying if more time or resources were allotted. You should also provide us with your comet.ml tracking for us to check the training process of your experiments. In essence, even if you do not achieve the results in the paper, as long as it shows that you have put in significant effort and thought into the assignment (as evidenced in the written portion), you will receive substantial partial credit.

The grades of the programming assignments together will make up 60% of the final grade. However, since the difficulty of the projects may vary, the weightings of the assignments will be adjusted accordingly (for instance, difficult projects in which a significant amounts of students struggle to achieve good results may have a lower weighting.) **Note, however, that you must complete all of the programming projects in order to receive credit for this course.**

Each of the programming projects is due by midnight on the date listed on the website. **You will have seven (7) free late days to use towards all but the last project, and each project may only have up to three (3) late days applied.** After your late days are expended for the project, you will lose 20% of your project grade for each extra day your handin is delayed.

Additionally, if you want a TA to regrade an assignment, you need to ask that TA within two weeks of receiving your assignment back.

Final Project

There will be a final project at the end of the semester. We encourage you to apply what you've learned in class to solve real-world problems or explore research topics of your interest. You can finish the final project either individually or in a group of at most 3 people. If you have more people in your group, we expect you to work on a more difficult project that requires more work. We will grade your final project according to your demo and presentation at the end of the semester. A more detailed guideline and grading rubric will be released on the course page. **No late day is allowed on the final project.**

Labs and TA Hours

This is a more exploratory class and we encourage you to interact more with the staff and other students. We want to encourage you to come to TA hours and make TA hours a collaborative space that the class can work together.

TA hours are offered over Zoom during the times posted on the course web page. To join hours, add yourself to the Signmeup queue and write a brief description of your question in the provided field. Once you are "claimed," join the Zoom link posted in the Signmeup queue. The TA may claim multiple students at a time; if you need individual attention, note

this in the question field.

Labs are not mandatory. They are there to help you learn certain tools and skills you may find useful when doing programming projects or in your Deep Learning career in general.

Participation

To get full marks on participation, you will need to come to at least 3 TA hours. You are welcome to talk about anything NLP or Deep Learning. We have an awesome TA staff and you will have a chance to get to know at least a few of them. Coming to TA hours is also a good way to meet and talk to other classmates and team up for the final project.

Coding

Since this is not a software engineering course, we will not be enforcing stringent style guidelines, but you should write so that someone unfamiliar with your code will be able to understand what your program is doing (add plenty of comments, break up code into smaller functions, etc.) If you turn in a partially-functional assignment and we are unable to tell what you were trying to do, we will have a much more difficult time giving partial credit.

As that translates to an official policy, so long as your code produces the expected output(s) and adheres to any specific project restrictions (runtime, etc.) then you will not lose points for poor design or coding practices. However, as this is not a software design course, it is not the responsibility of the TAs to attempt to understand the intentions underlying confusing code.

It is in your interest to become proficient in PyTorch: some research groups use it, and since the documentation for PyTorch is more difficult to understand than TensorFlow's, it is better to learn it in a course setting than at the last minute when you need it for research. Consequently, we provide bare stencils for the assignments in PyTorch, and we provide a lab to help you become familiar with it. However, you are free to use TensorFlow instead if you prefer, and you may modify the stencils as you see fit; however, please use the same command-line interface and options provided in the stencils.

Extensions / Incompletes

Requests for extensions on assignments or incompletes for the course should be directed to Professor Charniak. **However, please note that he will be very reluctant to grant an extension or incomplete even with a dean's note.**

Collaboration Policy

In general, we strongly encourage discussion and learning from your peers. However, ultimately, you must be able to produce original work, and the work that you submit must be your own.

- Reflecting the exploratory nature of the assignments, discussion of the readings is permitted and highly encouraged.
- You are allowed to discuss implementation ideas, such as the neural network design, network hyperparameters, or methods of preprocessing data. Our TA hours will be designed to facilitate such an environment. **However, we ask that in the written portion of the assignment, you include the names of those you have received significant assistance from.** This is akin to acknowledging a reference in a research paper. Furthermore, while sharing of ideas is encouraged, make sure that you are at least doing some original work of your own, and not solely relying on the ideas of others.
- While we encourage sharing of ideas, **the actual code, writeup and handin *must be of your own work*.** Do not upload your solutions online, including on public Github repositories. You cannot collaborate on the actual code details with other people, nor should you refer to online codebases. Your handin is subject to MOSS testing to enforce this rule.
- To save time in debugging, we will allow students to help debug each other's code. **However, again, please list the people that you received debugging help in the written part.** However, you must take care that you do not recall specific code details from the person you are helping. In particular, have your laptop or code window closed while helping debug someone else's code. Do not take away any notes while debugging someone else's code. After debugging help, take a break before taking a look at your own code.

Academic dishonesty will not be tolerated. This includes cheating, lying about course matters, plagiarism, or helping others commit a violation. Plagiarism includes using the work of others without a citation. Students are reminded of the obligations and expectations associated with the Brown Academic and Student Conduct Codes. This **will** be enforced.

Diversity & Inclusion

Our intent is that this course provides a welcoming environment for all students who satisfy the prerequisites. Our TAs have undergone training in diversity and inclusion, and all members of the CS community, including faculty and staff, are expected to treat one another in a professional manner. If you feel you have not been treated in a professional manner by any of the course staff, please contact either the instructor, [Ugur Cetintemel](#) (Dept. Chair),

Tom Doeppner (Vice Chair) or Laura Dobler (diversity & inclusion staff member). We will take all complaints about unprofessional behavior seriously. Prof. Krishnamurthi has good notes on this area. To access student support services and resources, and to learn more about diversity and inclusion in CS, please visit [this webpage](#).

Brown welcomes students from all around the country and the world, and their unique perspectives enrich our learning community. To empower students whose first language is not English, an array of support is available on campus, including language and culture workshops and individual appointments. For more information, contact the English Language Learning Specialists at ellwriting@brown.edu.

Accommodations

Brown University is committed to full inclusion of all students. Please inform the professor if you have a disability or other condition that might require accommodations or modification of any of these course procedures. You may contact through email, come to office hours, or speak individually after class, and your confidentiality will be respected. We will do whatever we can to support accommodations recommended by SEAS. For more information contact Student and Employee Accessibility Services (SEAS) at 401-863-9588 or SEAS@brown.edu. Students in need of short-term academic advice or support can contact one of the deans in the Dean of the College office.

Mental Health

Being a student can be very stressful. If you feel you are under too much pressure or there are psychological issues that are keeping you from performing well at Brown, we encourage you to contact Browns Counseling and Psychological Services CAPS. They provide confidential counseling and can provide notes supporting extensions on assignments for health reasons.