## Course Prefix and Number: CS4650

Course Name: Natural Language

Section Number: A

Credit Hours: 3

Semester and Academic Year: Fall 2026

## Instructor Information

**Instructor: Wei Xu**

**Email: wei.xu@cc.gatech.edu**

## General Course Information

### Description

This advanced undergraduate AI course provides a mathematically grounded introduction to modern data-driven and deep learning approaches for natural language processing (NLP), with an emphasis on likelihood-based objectives, gradient-based optimization, and generalization. **The course requires strong mathematical and programming skills, in-person attendance, and has an in-person midterm exam.**

Topics include linguistic structure (e.g., part-of-speech, syntax, and semantics); classical machine learning models (e.g., logistic regression, hidden Markov models, and conditional random fields) with derivations of their objective functions, independence assumptions, and dynamic programming algorithms for exact and approximate inference (e.g., forward–backward, Viterbi algorithms); and neural architectures including feedforward, recurrent, and convolutional neural networks, attention mechanisms, and Transformers.

The course further covers structured prediction, sequence-to-sequence modeling, decoding algorithms (e.g., beam search), and modern approaches to training and adapting large language models, including pretraining objectives, fine-tuning, and reinforcement learning from human feedback. Assignments emphasize implementing models from first principles, deriving key algorithms, and building neural architectures in PyTorch for tasks such as text classification, named entity recognition, and text generation.

## Pre-Requisites

This is an advanced AI course on Natural Language Processing. Modern NLP is heavily based on Machine Learning and Deep Learning. To succeed in this class, you will need a very strong math and programming background that can sufficiently make you feel at ease in the machine learning class (CS 4641/764). **Students should have completed CS 4641/7641 in a prior semester; taking it concurrently is not sufficient preparation. The course assumes mastery of mathematical concepts in probability, linear algebra, and multivariable calculus**. You should also be comfortable working on medium-size software projects, learning and using new libraries (in particular, PyTorch) quickly, and **debugging complex code when error messages are incomplete or absent, including writing your own unit tests, inspecting intermediate variables, and systematically isolating and diagnosing issues, etc.**

There will be a math background test (due in the 1st week of the semester before registration closes) and a warmup programming assignment (implementing logistic regression algorithm and gradient descent from scratch, due in the 2nd week). If you find these difficult, we recommend waiting to take this class in a later semester, once you are better prepared. Please reach out to the course staff to discuss whether you have the right background to succeed in this course, especially if there are any symbols or concepts on these assignments which you are unfamiliar with.

## Course Learning Outcomes

- Formulate natural language processing problems using probabilistic and mathematical frameworks
- Understand and implement core algorithms for classical NLP models (e.g., logistic regression, hidden Markov models, conditional random fields), including dynamic programming methods such as forward–backward and Viterbi.
- Analyze and implement neural architectures for NLP (e.g., RNNs, CNNs, attention mechanisms, Transformers), including understanding their parameterization, training objectives, and optimization behavior.
- Design, train, and evaluate NLP systems for tasks such as text classification, named entity recognition, and sequence-to-sequence generation using modern toolkits (e.g., PyTorch).

- Critically assess model performance and generalization, and apply appropriate techniques for training and adapting large language models (e.g., fine-tuning, RLHF) to real-world applications.

### Required Course Materials

There are two excellent NLP textbooks that are freely available online. Readings will be assigned from both. There is value in seeing multiple perspectives on the same material. If a concept you encounter seems confusing at first, try reading about it in the other book to get a different perspective.

(1) "Speech and Language Processing" (3rd Edition) by Dan Jurafsky and James H. Martin. https://web.stanford.edu/~jurafsky/slp3/
(2) "Natural Language Processing" by Jacob Eisenstein. https://github.com/jacobeisenstein/gt-nlp-class/blob/master/notes/eisenstein-nlp-notes.pdf
(3) There will be other assigned readings as well, most of which are research papers and blog posts.

### Grading Policy:

- Programming Assignments: 28%
- Written Assignments: 20%
- In-Class Midterm Exam: 20%
- Final Project: 25%
- Attendance, In-Class Presentation, and Participation: 7%

Graded homework assignments will include both written and programming assignments. Assignments should be submitted to Gradescope by 11:59pm on the day they are due. All assignments are individual work (only exception is course project). Please email your homework to the instructor in case of any technical issues with submission.

Each student will have 6 flexible days to turn in late homework throughout the semester. The late days will be applied to homework assignments in the order of submission; **no late days can be applied to the final project** (a group assignment). As an example, you could turn in the first homework 2 days late and the second homework 3 days late without any penalty. After that you will lose 5% for each day an assignment is handed in late. These 6 late days are meant for personal emergencies; if you use late days for non-emergency situations but later encounter emergencies in the semester, you will not be given extra late days. No late days will be allowed for the final project, due to the tight deadline for posting the final grades as required by the university. Late days will be counted in 24-hour chunks to the entire homework and rounded up, i.e., if you are late for 2 hours for a homework assignment, that will count as 1 late day.

All homework will be rescaled proportionally into the final numerical grade, which will then be mapped to letter grade according to a cutoff based on the overall class grade distribution. **The standard cutoff is 90/80/70% for A/B/C, but we may curve up** (never down), i.e., use lower cutoffs than these. The cutoffs will only be determined after we grade the final project at the end of the semester. Students must obtain at least 50% overall grades to pass the class; students who choose the pass/fail option shall not participate in the final group project.

## Description of Graded Components

### In-Class Midterm Exam

The midterm will cover topics that are selected from the lectures, assigned reading, and homework assignments. The midterm will follow a format similar to the written assignments. **It will be administered in person (no remote option) and will be closed-book and closed-notes.**

There will be no make-up exam, except for students who have an official approved absence from the Office of the Dean of Students. We will set the exact date for the midterm exam one month before the exam based on the class progress.

### Programming Assignments and Written Assignments

We plan to assign four programming assignments (including a programming background test) that provide hands-on experience implementing algorithms discussed during lecture. The assignments are in Python, and make use of Numpy and Pytorch. These will require non-trivial computation to complete; we recommend using Google's Colab platform which provides easy access to GPUs. Completing these projects will require waiting for your models to train (this can range from about 30 minutes to hours depending on the efficiency of your implementation), so we strongly recommend starting work on these programming assignments well in advance of the deadline. If you start working on an assignment the day before it is due, it is unlikely you will be able to complete it on time.

The written assignments will mostly be mathematical/computational. You can complete the homework in text editors, or alternatively, scan and upload your solution. Please write answers clearly, since we won't be able to provide credit for answers that we are not legible.

### Final Project

The final project is an open-ended, student-directed assignment, with the goal of gaining experience applying the techniques presented in class to real-world datasets. Students should work in groups of 2–4. Students who wish to work individually (group 1) must submit a written project proposal and obtain instructor approval prior to the midterm. It is a good idea to discuss your planned project with the instructor to get feedback. The final project report should be 4-5 pages (excluding references). The report should describe the problem you are solving, what data is being used, the proposed technique you are applying in addition to what baseline is used to compare against.

The grading rubric for the final project is as follows:

- Clarity (1-5) For the reasonably well-prepared reader, is it clear what was done and why? Is the report well-written and well structured?
- Originality / Innovativeness (1-5) How original is the approach? Does this project break new ground in topic, methodology, or content? How exciting and innovative is the work that it describes?
- Soundness / Correctness (1-5) First, is the technical approach sound and well-chosen? Second, can one trust the claims of the report – are they supported by proper experiments, proofs, or other argumentation?
- Meaningful Comparison (1-5) Does the author make clear where the problems and methods sit with respect to existing literature? Are any experimental results meaningfully compared with the best prior approaches?
- Substance (1-5) Does this project have enough substance, or would it benefit from more ideas or results?  Note that this question mainly concerns the amount of work; its quality is evaluated in other categories.
- Overall (1-5) This will be the final score based on the overall quality of the project. This is not the sum of the above aspect-based scores.

### Attendance, In-Class Presentation, and Participation

The class requires in-person attendance. Attendance will be taken on randomly selected class days. Students may miss up to two such sessions without penalty; additional absences will result in a grade deduction, except for official approved absence from the Office of the Dean of Students. Attendance may be recorded using various methods, including roll call, sign-in sheets, in-class quizzes, or student card readers.

Depending on class size and scheduling, each student may be asked to give a short in-class presentation on a scheduled date. Presentation dates will be announced at least one month in advance.

Students will also receive credit for asking and answering thoughtful questions related to the course content on Piazza, engaging in discussion in class and generally for participating in the class. There are many ways to show participation. Asking a question that is marked as a "good question" by an instructor on Piazza, or having an answer that is marked as an "endorsed answer" is one example. Asking insightful questions, and engaging in discussion during class is another example. Please be polite and respectful towards TAs and other students in the class.

## Course Policies

### Attendance and/or Participation

This course requires in-person attendance. Attendance will be taken on randomly selected class days. For absences due to illness or personal emergencies, students shall submit an absence request and obtain approval through the Office of the Dean of Students.

### Academic Integrity

Georgia Tech aims to cultivate a community based on trust, academic integrity, and honor. Students are expected to act according to the highest ethical standards. Review Georgia Tech's Honor Code and the student Code of Conduct.

Any student suspected of cheating or plagiarism on a quiz, exam, or assignment will be reported to the Office of Student Integrity, who will investigate the incident and identify the appropriate penalty for violations.

### Core IMPACTS

Core IMPACTS is the University System of Georgia's General Education curriculum. If you are teaching a course that counts towards Core IMPACTS, you should include a syllabus statement about the Core area and associated career competencies. This resource developed by the Center for Excellence in Teaching and Learning and Online Education at Georgia State University includes template syllabus statements for each of the Core IMPACTS areas that you may adapt for your course.

### Accommodations for Students with Disabilities

If you are a student with learning needs that require special accommodation, contact the Office of Disability Services (404-894-2563) as soon as possible to make an appointment to discuss your special needs and to obtain an accommodations letter. Please also e-mail me as soon as possible in order to set up a time to discuss your learning needs.

### Student-Faculty Expectations Agreement

At Georgia Tech, we believe that it is important to strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty members and the student body. The Student-Faculty Expectations articulate some basic expectations that you can have of me and that I have of you. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. Therefore, I encourage you to remain committed to the ideals of Georgia Tech while in this class.