Natural Language Processing (CS 7650)

Fall 2025

Instructor: Mark Riedl (riedl@cc.gatech.edu)

General Course Overview

Natural Language Processing (NLP) seeks to endow computers with the ability to intelligently process human language. NLP components are used in conversational agents and other systems that engage in dialogue with humans, automatic translation between human languages, automatic answering of questions using large text collections, the extraction of structured information from text, tools that help human authors, and many, many more.

This course will teach you the fundamental ideas used in key NLP components as well as current state-of-the-art practice in developing NLP algorithms.

Prerequisites

Students are expected to be proficient in the python programming language. Course requirements include programming assignments written in the python programming language using Jupyter notebooks on a personal computer or on Google Colab (or similar cloud development environment). You will not need a GPU-capable environment for the assignments.

Students are expected to have background in data structures and have had some exposure to computational complexity (e.g., analysis of algorithm complexity, finite automata). Additionally, students are expected to have background in basic probability, linear algebra, and calculus. Students are encouraged to have taken a course in artificial intelligence or machine learning. Of relevance are familiarity with linear classifiers, perceptrons, naive Bayes, and logistic regression. While helpful, it is not assumed.

Course Materials and Resources

Online Textbook: Natural Language Processing (2018) by Jacob Eisenstein.

Computing Environment: Programming assignments will be done in Python using Jupyter Notebooks. Assignments can be performed on personal computers with sufficiently modern CPUs. Students may benefit from NVIDIA GPUs capable of running CUDA, however we find that it is not strictly required. Google Colab is a cloud environment that runs Jupyter Notebooks and can access free GPUs. The free tier of Colab has service limits, and a Google Colab Pro account, which has a small monthly cost, provides access to more RAM and has higher caps on GPU usage. Students may explore other comparable services, though we have not investigated the practicality of other services.

Canvas: Course materials, assignment downloads, quizzes, and exams.

Ed Discussions. For Official Announcements, and Forums for discussion. All class discussions will be on the Ed Discussion site. Here are some very specific guidelines for these discussions, which must be adhered to:

- All posts must be professional and cordial and about/related to the course material at hand.
- Students WILL NOT post specific answers to any of the assignments to Ed Discussion before the due date of said assignments unless the TAs request it in a PRIVATE post.
- Before asking a question on the Forum, students should search for an answer to their question. It most probably has been discussed already.
- Instructor team will attempt to answer all questions, as soon possible. But please do NOT expect immediate responses. TAs are humans too with courses and other responsibilities. TAs are instructed let students answer each other's questions too, as that support more interactive learning. It is always wise to start programming assignments early, and to help others out when you can. Set the example you hope to benefit from later.
- Students can post anonymously to the class, but their IDENTITIES will be known by the instructor team.
- Instructor team is required to maintain privacy of all students, so please ensure that you communicate with them privately (using the private channels via Ed Discussion) to communicate with them.
- If there is a complaint about the class, please DO NOT post a public note to Ed Discussion. Please communicate directly with the instructor team. We will do our best to address it. If it is NOT addressed, please use OMS Assistance.

Gradescope. Students will download assignment instructions and materials from Canvas and submit their solutions to Gradescope for grading.

No information will be shared via any other site (Facebook, etc.). Students are welcome to create their own social media sites, but none of the instructors are required to be on those sites and will not participate there regularly.

Grading

- **Quizzes:** 10%
- **Programming Assignments:** There will be 6 programming assignments:
    - 1: Introduction to neural networks (5%)
    - 2: Classification (10%)
    - 3a: Language Modeling, part a (10%)
    - 3b: Language Modeling, part b (10%)
    - 4: Distributional Semantics (10%)
    - 5: Mini-project (15%)
- **Final Exam:** There will be one final, comprehensive exam (15%)
- **Midterm Exam:** In lieu of a traditional exam, you will be asked to review a conference paper from an NLP conference within the last ten years (15%)

Programming Assignments

The primary goal of this class is to provide hands-on learning experiences building natural language processing systems. We have broken these experiences into 6 programming assignments.

Programming assignments will be conducted in python using Jupyter notebooks. You may use Google Colab, which natively supports Jupyter. You may also run the notebooks locally on your own machine. If you run the code on your own machine, you should use Jupyter Lab version 4.0 or greater.

The first assignment will familiarize you with the programming environment and Pytorch API (a python library for building and training neural networks).

Assignments 2, 3a, 3b, and 4 will walk you through the construction of increasingly more sophisticated natural language processing models, most of which will be based on neural networks. This will largely involve filling in the code for pre-defined functions.

The final programming assignment, the mini-project, will ask you to work with preparing a pre-determined dataset and building, training, and testing a specified type of natural language processing model from scratch. The notebook will walk you through the steps but will not pre-define functions to be completed.

All programming assignments except the last will be graded via autograder on Gradescope. The autograder will be provided as part of the notebook so that you can perform self-assessment. Grades will be determined by the same autograder. Any attempt to modify the autograder will result in an automatic zero. The mini-project programming assignment will be manually graded via code inspection. There will be written components in which some analysis and textual description of the solution is given; this will likewise be manually graded.

Programming assignments will be submitted to Gradescope. **Programming assignment notebooks should have notebook cell outputs saved**.

Submissions to Gradescope must run to completion without error. It is possible that there will be differences between your local computing environment and Gradescope's computing environment due to, for example, default dependencies on your local machine that are not present on the gradescope machine. Also, if GPUs are used, different GPUs on local machines versus gradescope machines can result in different results. You should always check your gradescope results. Gradescope returns a status as to whether it is able to completely process the notebook or not. A "green light" may not indicate that you are getting the same results as any run on your local computing environment. A "green light" may also occur even if the autograder times out because that is not a crash.

Midterm Exam

The midterm exam will be delivered on Canvas and proctored by HonorLock. It will be a timed test (1.5 hours) with multiple-choice and short answer questions. It will be open-notes, closed book, and closed internet (unless we choose to whitelist websites that you will need to answer a question). You will be able to upload and access a PDF version of your notes.

Final Exam

The final exam will be delivered on Canvas and proctored by HonorLock. It will be a timed test (3 hours) with multiple-choice and short answer questions. It will be open-notes, closed book, and closed internet (unless we choose to whitelist websites that you will need to answer a question). You will be able to upload and access a PDF version of your notes.

Quizzes

Quizzes act as attention tests at the end of each module. Quizzes will involve multi-choice questions about the lecture materials and will be delivered via Canvas. The Quizzes will all together total 10% of the grade.

Grading Scale

Grading Scale (for each assignment/unit and for the entire class).
A: At or above 90%
B: 80%-89.99%
C: 70%-79.99%
D: 60%-69.99%
F: Below 60%

Late Day Policy

Over the course of the semester, you'll have 5 “free” late days to submit programming assignments (except the final mini project). A late day is used one minute after the due date. A second late day is used 24 hours after that, and so on. Late days are determined by Gradescope submission timestamps. Late days **cannot** be used on quizzes or exams or the final mini-project. Our intention is to give you some flexibility around your work commitments, family obligations, vacations, and the like.

Additional rules:

- For every extra late day (past the 5 “free” late days) used, you'll incur a 25% deduction.
- If you have a medical or family emergency, please contact the Dean of Students who may grant an exception to the late policy if your circumstances warrant it. We must receive approval from the Dean to grant an exception.

Regrading Policy

Regrade requests can be made via Gradescope. Please provide clear details as to why you are requesting a regrade. All regrade requests must be made within **ONE (1) week** of the grade

release. For grades released in the last week of the term, the regrade request must be made by the last day of the final exam week.

Due Dates

All due dates will be on Canvas, and the time zone will be Anywhere on Earth Time (AoE) time. Please plan accordingly, especially around Daylight Savings changes in the US and in your location.

Honor Code

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. The Georgia Tech Academic Honor Code applies to all aspects of this course. Plagiarism is a violation of the Academic Honor code. To plagiarize is defined by Webster's as "to steal and pass off (the ideas or words of another) as one's own; use (another's production) without crediting the source." Any student suspected of cheating or plagiarizing 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. For any questions involving these or any other Academic Honor Code issues, please consult us or http://catalog.gatech.edu/policies/honor-code/.

Collaboration Policy

Collaboration between students on work assigned in class is fine. You are encouraged to discuss your work with each other. But each individual students MUST submit their own work, done solely by themselves. In some cases, you may have had a fellow student or a non-student friend, help you with an assignment. You are REQUIRED to acknowledge any help you may have received in completing the work assigned, even as small as suggesting a possible path to a solution. Please be explicit and provide details. We will be checking for code plagiarism in our assessment, so please NO copying code from the Web/Internet. Any code snippets must be cited and limited to a maximum of 3 lines. We understand you may not be familiar with some libraries and APIs presented in this class and you will likely look up usage examples for individual functions. You may study these examples, but the code used in your assignment must be your own. Document your web/internet sources so that if our plagiarism detector is triggered we can determine if it is a false alarm. Some websites are expressly forbidden: you may not use any information from Chegg, CourseHero, Cliffsnotes, or any other website that is expressly set up to "help" students do their homework by providing solutions or connect you to people who assist. If you are uncertain, please ask. If you find yourself tempted, then you should be talking to the TAs. They are a free and helpful and misconduct-free resource.

To protect yourself and to protect others we recommend the following heuristics when communicating with others about course assignments:

1. Do not copy and paste your own code to share with someone else. If they in turn copy your code into their assignment, this will trigger plagiarism detectors. You may want to write out your suggestions in English (or your non-coding language of choice).
2. Do not copy more than 3 lines of any code that you find on the internet. When more than one person in a class copies the same piece of code from the internet, this triggers plagiarism detectors. Instead, use this as a learning episode: close your code, study the example, close down the example, and try to write it yourself. You will find this more fulfilling in the long run than copying, even if it is less than 3 lines. Document you use of internet sources by adding a comment or markdown cells to your notebooks. There is no shame in admitting that you had to go to stackoverflow. There is nothing we would like more than to clear you of false positives when our plagiarism detector flags multiple people referencing at the same stackoverflow discussion.
3. Do not share your screen when your Google Colab notebook is visible so that others can see, record, and screen capture.
4. If communicating in person or via video conferencing, use the "whiteboard policy": write by hand (if possible) on a whiteboard app and erase the whiteboard afterward. Do not take a photo or screen capture. Only write in a human language, never code. For example:

```
Iterate through each point in the training data set:
    Covert each data point into a vector.
    Input your vectorized data point into the model.
```

Deviating from these heuristics does not automatically qualify as academic misconduct; however, following these heuristics greatly reduces the probability that your collaboration will not cross the line into misconduct.

As part of this course's grading process, any **suspicion** of copying WILL be reported to the Office of Student Integrity for further analysis.

All students must also ensure that they DO NOT make any of the code for problem sets publicly available and are required to take steps to prevent future students from having access to it. Consequently, if you're using any version control systems such as git, please make sure that you mark your repositories as private.

You may not collaborate at all on the exams or quizzes. Students are not to discuss any questions or answers from the exams with classmates or anyone else until after the testing period is complete.

Course Materials are Copyrighted

We do not allow course materials—videos, transcripts—to be publicly disseminated. All materials and assignments are copyrighted by Georgia Tech. You have permission to use them for the purpose of the class. Making videos, screenshots, or transcripts publicly available may

trigger Georgia Tech to pursue legal copyright protection actions. Some of the course material has been made publicly available by the instructor in the form of blogs. You may link to these materials but not distribute them further without instructor permission.

You are prohibited from sending screenshots, video, or transcripts to a cloud-hosted LLM such as ChatGPT, Claude, Microsoft Copilot, etc. These services may use any prompts or material transmitted to them as part of future training. The inclusion of copyrighted material in prompts is thus considered an unauthorized copy and dissemination of materials to an unauthorized recipient.

You **may** share your notes publicly and with other members of the class. A word of warning, however: the reason one takes notes is to engage mental resources with the course material. The process of taking notes itself is self-reinforcing. One must listen, understand, decide what is important enough to write down, figure out how to summarize and lay out your notes. Note taking ensures that you are engaged in active-listening instead of passive-listening. Relying on other people to take notes for you is not as effective a learning processes as taking notes oneself.

Use of ChatGPT and other Large Language Models

Use of AI as a coding assistant or to assist with the answering of exam questions or quiz questions or to generate code to be turned in as part of an assignment is strictly **prohibited**. The purpose of this class is to help you build cognitive knowledge skills and reinforce coding skills. We are aware that some of our assignments will ask you to implement standard solutions like an LSTM. We are aware that there are packages that make it so one will never have to do this in real life. We also believe that one does not understand a system nearly as well as when one must work out the details by hand. You will get down into the weeds of neural networks and probability computations. You will understand what neural networks are doing in a way that you cannot learn by reading and watching lectures. The coding assignments were carefully chosen to reinforce your cognitive understanding. Coding is like lifting weights. Using AI to generate code for a class is like using a forklift to move the weights. Those weights do not need to be moved from one place to another. The change is happening inside you. No one regrets being fit. One might regret never being fit, or losing that fitness. To use a crafting metaphor: you learn crafts by following patterns at first. No one needs another pair of knit socks (I have some, thanks). But you experience all the little things that can go wrong that aren't in the tutorials, and learn how to correct for them so that when you want to make something completely new, you will not flounder and fail. The same applies to coding assignments. They may not always be fun. But if AI is doing the coding for you, you may feel like you are learning by reading the generated code, but that is not how learning works. You are not learning in nearly the same way as having to think through the creation of every step of the algorithm. The little differences between lecture and working code are there for you to figure out. Problem solving means cognitive growth.

Your development environment software likely has an AI option. Google Colab has an AI coding assistant built in, for example. I recommend that you disable or turn them off or ignore them.

You may want to ask questions about lecture material. We do recommend that you post to the discussion bulletin board. The instructional team would rather have you ask us than an AI. But if you do, make sure you:

1. Ask questions that do not involve any code. You may receive code anyway, in which case do not copy from the AI output. You might want to ask that code not be given in answers.
2. Never hit "Copy" within your conversation with an AI assistant. If you copy code from a LLM, you are edging up against academic misconduct. If you copy your code into the LLM, it will generate code in return, so you will likely end up using the LLM's code, and now you have to resist copying the LLM's code response. Instead, use your interaction with the AI assistant as a learning experience, then let your assignment reflect your improved understanding.
3. Do not have your assignment and the AI agent open at the same time. Like above, use your conversation with the AI as a learning experience, then close the interaction down, open your assignment, and let your assignment reflect your revised knowledge. This heuristic includes avoiding using AI directly integrated into your composition environment: just as you should not let a classmate write content or code directly into your submission, so also you should avoid using tools that directly add content to your submission.
4. In the likely event that your code composition environment integrates language-to-code generation capabilities or code autocompletion, it is recommended that you deactivate this functionality. The reason why this is important is because LLMs have a particular style and preference for how to write code. This will trigger our plagiarism detectors because when several people use an LLM they can get nearly identical code, even if they have never met. You may never have to write some pieces of code from scratch, but you will learn more by doing it yourself at least once.

Deviating from these heuristics does not automatically qualify as academic misconduct, and following these heuristics does not automatically make you free of the possibility of being flagged for academic misconduct.

AI usage has been endemic. We will use a variety of techniques in the course to flag the potential use of AI in homework, exam, and quizzes, including plagiarism detectors. People using the same AI for the same assignments will often produce identical code. We now have more than 1,000 reference solutions and flags are very high confidence. Even though there are only a few ways to solve any particular problem and a lot of guidance is given in the instructions, individual coding signatures are still very distinct.

Accommodations for Students with Disabilities

If you have learning needs that require special accommodation, contact the Office of Disability Services at 404-894-2563 or http://disabilityservices.gatech.edu/ as soon as possible to make an appointment and discuss your special needs and to obtain an accommodations letter. Please also talk with us to discuss your learning needs.

Student-Faculty Expectations Agreement

At Georgia Tech, it is important to strive for an atmosphere of mutual respect, acknowledgement, and responsibility between faculty member and the student body. Please see http://catalog.gatech.edu/rules/22/ for an articulation of some basic expectations that you can have of us and that we have of you. These were adopted by both the faculty senate and the student government. In the end, simple respect for knowledge, hard work, and cordial interactions will help build the environment we seek. Therefore, we encourage you to remain committed to the ideals of Georgia Tech while in this class, especially during class discussion.

Inclusion

The Georgia Institute of Technology is committed to creating a campus free of discrimination on the basis of race, color, religion, sex, national origin, age, disability, sexual orientation, gender identity, or veteran status. We further affirm the importance of cultivating an intellectual climate that allows us to better understand the similarities and differences of those who constitute the Georgia Tech community, as well as the necessity of working against inequalities that may also manifest here as they do in the broader society.

If you have questions

We'll be using Ed Discussions in this course. Please use that forum to post questions and comments. The instructional staff and other students will be of assistance there. The instructors may offer live office hours at certain milestones during the semester.

If after contacting your TA and the instructor you do not feel your issue has been resolved, you may escalate the issue by emailing oms-advising@cc.gatech.edu.

Schedule

The following schedule is a suggested timeline for engaging with modules, though module content can be accessed at any time. The due dates below are for planning purposes and are subject to change. Official due dates will be on Canvas. Students are responsible for monitoring Canvas and Ed Discussion for announcements about due dates.

Week	Date	Module	Reading	Released	Due
1	Aug 18	1: Intro to NLP 2: Foundations	Ch 1 Appendix A	Aug 18: • Onboarding quiz • Module 1 quiz • Module 2 quiz Aug 22 • HW1 (neural nets)	Aug 24: • Onboarding quiz

Week	Date	Module	Reading	Released	Due
2	Aug 25	3: Classification	Ch 2, 3	Aug 25: • Module 3 quiz	Aug 31: • HW1 • Module 1 quiz • Module 2 quiz • Module 3 quiz
3	Sept 1	Language models	Ch 6	Sep 1: • Module 4 quiz Sep 5: • HW2 (classification)	
4	Sept 8	4: Language models	Ch 6		Sep 14: Module 4 quiz
5	Sept 15	5: Semantics	Ch 14	Sep 15: • Module 5 quiz Sep 19: HW 3a (language models)	Sep 21: • HW 2 • Module 5 quiz
6	Sept 22	6: Modern Neural architectures		Sep 22: • Module 6 quiz	
7	Sept 29	6: Modern Neural architectures		Oct 3: • HW 3b (language models)	Oct 5: • HW 3a • Module 6 quiz
8	Oct 6	7: Information retrieval		Oct 6: • Module 7 quiz Oct 8: • Midterm exam	Oct 12: • Module 7 quiz • Midterm exam
9	Oct 13	Fall Break			
10	Oct 20	8: Task-oriented dialogue	Ch 19	Oct 20: • Module 8 quiz Oct 24: • HW 4 (distr. Semantics)	Oct 26: • Module 8 quiz
11	Oct 27	9: Summarization	Ch 11	Oct 27: • Module 9 quiz	Nov 2: • Module 9 quiz • HW 3b
12	Nov 3	10: Machine Reading		Nov 3: • Module 10 quiz Nov 7: • HW 5 (mini-project)	Nov 9: • Module 10 quiz
13	Nov 10	11: Open-Domain Question-Answering	Ch 18	Nov 10: Module 11 quiz	Nov 16: • HW 4 • Module 11 quiz
14	Nov 17	12: Machine Translation		Nov 17: • Module 12 quiz	Nov 23: • Module 12 quiz
15	Nov 24	13: Privacy-Preserving NLP		Nov 24: Module 13 quiz	Dec 1: • Module 13 quiz Mini-project
16	Dec 1	14: Responsible AI Last day of class First half week of exams		Dec 1: • Module 14 quiz • Final Exam	Dec 7: • Module 14 quiz • Final Exam
17	Dec 8	Final exam week			

All due dates are at midnight anywhere on Earth (AoE). For example, HW1 is due at 11:59pm AoE on Sunday, August 31st which is 7:00 AM Atlanta-time (EST) on Monday September 1st. But don't worry, Canvas adjusts the due date based on your computer's locale - be sure your computer locale is set correctly.

Note that most deadlines are on Sundays AoE, but some deadlines are mid-week. We recognize that many OMS students work jobs and try to ensure that all assignments span weekends.