CS 4740: Natural Language Processing, Spring 2026 (undergraduate)

Cross-listed as COGST 4740 / LING 4474 / CS 5740

Instructors: Tanya Goyal

TAs: Wayne Chen, Son Tran, Chengyu Huang, Aileen Huang, Anand Bannerji, Andrew Hu, Jeffrey Huang, Frank Yang, Jay Talwar, Brianna Liu, Deniz Boloni-Turgut, Yunoo Kim, Mahitha Penmetsa

Course Administrative Assistant: Amy Finch Elser (email: ahf42@cornell.edu)

Lecture: Monday 7:30PM-10:00PM, Statler Hall 185-Aud

Office Hours and communication with staff:

Instructor Office Hours — 1 -2 p.m., Tuesday, Gates 441A or by appointment (email: tanyagoyal@cornell.edu)

TA Office Hours — (will start from Jan 27, 2025 onwards, timings and rooms TBD)

[Link to course syllabus and schedule here]

Description

This course is an introduction to modern natural language processing (NLP). Today, NLP is at the heart of many exciting technologies including the widely popular large language models (LLMs) like ChatGPT and Claude. This course will lay the foundation for understanding how frontier LLMs like ChatGPT are built. We will cover traditional NLP approaches to langauge modeling, such as N-gram models to RNNs, followed by transformer model architectures that form the backbone of modern LLMs. This course will cover training recipes, data considerations and other ingredients that contribute to building these powerful systems. We will also cover topics related to factualily, retrieval-augmented language models, efficiency, etc. At the end of this course, students should be in a position to understand and critique recent research in NLP.

Resources

1. **Schedule:** The course schedule (including lectures materials and assignments) is provided here.
2. **Textbooks:** We will follow **Jurafsky and Martin, Speech and Language Processing, 3rd edition (draft).** Free online version is available here.

3. **Assignments:** You will submit assignments using **Gradescope.** For coding parts of the assignments, you will use colab.

Prerequisites

1. Strong programming skills are important. Three semesters of programming classes are strongly recommended (e.g., completion of CS3110). CS2110 may suffice if you individually could have successfully and easily completed the assignments by yourself.
2. Python experience. Pytorch experience (as through CS4780) not required but some students report it being very helpful.
3. Comfort with elementary probability.
4. Clear understanding of matrix and vector operations.
5. Familiarity with differentiation.

You will be asked to complete HW0 (see the schedule page) to test these prerequisites. If you find yourself struggling with HW0, please talk to the course staff to discuss if this course is appropriate for you.

Course Policies

Note: We reserve the right to make necessary changes to any policy on this page if it would jeopardize the smooth running of the course. We aim to avoid making alterations, and will try to be as transparent as possible about key changes, e.g., by posting to Ed Discussions.

Grading:

1. **HW0 Review Assignment (ungraded):** This is designed to test whether you have the necessary pre-requisite knowledge for this course. You should do this assignment individually and it should take you less than 2.5 hours to complete. We will not grade this assignment but highly encourage you to use this to test your readiness for the course**.** If you have difficulty completing any part of this assignment, please contact the course staff to discuss whether this course is appropriate for you.
2. **Assignments (60%):** This course will consist of 4 take-home assignments (with possible milestones).

   Each of the 4 assignments will account for **15%** of the course grade. The assignments will be a mix of coding components and non-coding conceptual questions. We expect these to take tens of hours each, so you should start in time and budget enough time for each assignment. You can do these assignments individually or in groups of 2; all students in a group will recieve the same grade. We strongly encourage you to do these assignments in a group.

   (Only CS 5750) Students enrolled in CS5740 must complete an additional component for each 4740 homework **individually.** These components are graded as "satisfactory", "borderline", and "unsatisfactory". If a student receives two "borderline"s or one "unsatisfactory" among the four homeworks, we reserve the right to lower the student's letter grade as computed for 4740 by the equivalent of a "level", for example, from a B to a B-.

3. **Exams (midterm 20%, final 20%)**: These will test individual conceptual knowledge. The non-coding components of the homework assignments will provide a blueprint of questions to expect in these exams. **To receive a C- or above in the course, students must receive at least a C- on both exams.**

Collaboration policies:

1. Groups of two are allowed on all assignments except the review assignment (HW0) and the CS5740 add-on assignments. You can partner will anyone in the class (irrespective of registration to the undergraduate or graduate level of the course, or letter grade or S/U enrollment) **but please discuss time and effort expectations with your prospective partner.**
2. You do not have to have the same partner on each assignment.
3. Until all students' submissions have been posted for the assignments, you must never consult any other groups' written submission or code in any form. You are allowed to discuss conceptual doubts (e.g. how does the viterbi algorithm work?) but all the code and written report you submit must be your own (or your groups'). Additionally, you must not copy paste solutions from external sources like Stack Overflow or ChatGPT. Using these resources for debugging is allowed.
4. Please refer to the policy on the use of Generative AI (e.g. chatGPT, CoPilot, Claude, etc.) at the bottom of this page.

Late Submission Policy:

1. **Each student is given 5 slip days** to use throughout the course for the homeworks. You are allowed to use a maximum of 2 slip days per homework assignment. An example use of slip days is: 2 slip days for HW1, 1 slip days for HW2, 2 slip days for HW3 and none for HW4.
2. After the slip days are exhausted, you will incur a penalty of 10% on your grade for each additional day beyond the deadline for the assignment, up to a maximum of 4 days. After 4 days, your submission will not be graded.
3. **We will count slip days for each student individually.** If you are working in a group, members may incur different penalties for late submission depending on their individual slip day balance. Please co-ordinate appropriately with your partner.
4. **You are not allowed to use slip days for milestones.** The milestones only account for a fraction of the total assignment grade. We have found that students generally regret using slip days for milestones. To avoid any negotiations towards the end of the semester, we are issuing a blanket ban on using slip days for milestones.
5. You do not need to use slip days if you are sick or for other extenuating circumstances. Please email the instructors for these cases.

Accommodations for Students with Disabilities:

Your access in this course is important to us. Please give us [the instructors, the administrative assistant] your Student Disability Services (SDS) accommodation letter early in the semester so that we have adequate time to arrange your approved academic accommodations. If you need an immediate

accommodation, please send an email message to the instructors (cardie@cs.cornell.edu, tanyagoyal@cornell.edu), administrative assistant (Amy Finch Elser; email: ahf42@cornell.edu) and SDS at sds_cu@cornell.edu.

Other misc items:

1. **Enrollment options other than letter grades:**
   a. **S/U enrollment**: We will grade all your assignments and exams as if you were taking this course for a letter grade. We will convert this final letter grade to S/U using this policy: C- of better → S, D+ or below → U.
   b. **Auditing:** Sitting in on lectures is fine as long as there are enough physical seats. Students auditing the course, either officially in Student Center or unofficially ("just sitting in"), should ***not*** submit any work, partner with officially registered non-auditors, take any exams, or join office hours if the lines are long (we need to conserve our grading and staff resources).
2. **Policy on the use of generative AI (e.g. ChaGPT, CoPilot, Claude, etc.):**

   (Borrowed from Dr. Kuan Fang's course, thanks!)

   The work you do consists of writing code and natural language descriptions. To some extent, the new crop of "generative AI" (GAI) tools can do both of these things for you. However, we require that the **vast majority of the intellectual work must be originated by you, not by GAI**. You may use GAI to look up helper functions, or to proofread your text, but clearly document how you used it.

   In this class, for every assignment and final project, you can choose between two options:

   Option 1: **Avoid all GAI tools.** Disable GitHub Copilot in your editor, do not ask chatbots any questions related to the assignment, etc. If you choose this option, you have nothing more to do.

   Option 2: **Use GAI tools with caution** and include a one-paragraph description of everything you used them for along with your writeup. This paragraph must:

   1. Link to exactly which tools you used and describe how you used each of them, for which parts of the work.
   2. Give at least one concrete example (e.g., generated code or Q&A output) that you think is particularly illustrative of the "help" you got from the tool.
   3. Describe any times when the tool was unhelpful, especially if it was wrong in a particularly hilarious way.
   4. Conclude with your current opinion about the strengths and weaknesses of the tools you used for real-world compiler implementation.

   Remember that you can pick whether to use GAI tools for every assignment, so using them on one set of tasks doesn't mean you have to keep using them forever.

   Below we provide some guidelines for what is / is not ok when using GAI for this class:

   1. Example of something that is **allowed**: You write the initial code / writeup. You then use GAI to debug the code / improve writing flow. You do not use the system's output to add extra content.
   2. Example of something that is **definitely not allowed**: You essentially use GAI to generate most of the code / writeup, even if you later post-edit and correct the output.

3. Example of something that is **OK but requires special treatment**: You start with procedure in 1. But, the GAI suggests good points that you hadn't thought of before, or makes you realize that a point you had made isn't quite right. You may include this new material, but follow the guidelines above to document the use.