Ling 4400: Introduction to Natural Language Processing

Course Number: LING-4400 (Spring, 2025)
Lecture: Tuesday and Thursday, 11:00 - 12:15
Location: ICC 205b

Instructor: Ethan Wilcox
Office: Poulton 248 (Office Hours: Tuesday, 10:00 – 11:00 am)
Email: ethan.wilcox@georgetown.edu
Website: https://wilcoxeg.github.io/

Teaching Assistant: Jessica Lin
Office Hours: TBD
Email: yl1290@georgetown.edu

Summary: This course will introduce students to the basics of Natural Language Processing (NLP), a field that combines linguistics and computer science to produce applications, such as generative AI, that are profoundly impacting our society. We will cover a range of topics that form the basis of these exciting technological advances and will provide students with a platform for future study and research in this area. We will learn to implement simple representations such as finite-state techniques, *n*-gram models, and basic parsing in the Python programming language. Previous knowledge of Python is not required, but students should be prepared to invest the necessary time and effort to become proficient over the course of the semester. Students who take this course will gain a thorough understanding of the fundamental methods used in natural language processing, along with an ability to assess the strengths and weaknesses of natural language technologies based on these methods.

This course fulfills the college's *Quantitative Reasoning and Data Literacy* (QRDL) requirement as part of the Core Curriculum.

Learning Outcomes:

- Students will learn how to write code in Python
- Students will learn how to debug Python code using a debugger
- Students will become familiar with the theoretical fundamentals of programming
- Students will become familiar with the fundamental techniques used in natural language processing, including string manipulation, finite state techniques, language modeling, and vector space models
- Students will gain practice thinking critically about the behavior, abilities, and limitations of current NLP tools, including large language models

Course Schedule

Week	Module	Readings, and Assignments Out & Due

Jan 6, 2025 (1 session, Jan 9th)	Introduction	1/9 Assignment 1 (installing Python)
Jan 13, 2025	Intro to Python, String Manipulation	1/16 Assignment 1 due 1/16 Assignment 2 (palindrome checker)
Jan 20, 2025	Tokenization, Regular Expressions	Jurafsky & Martin, Chapter 2.1 - 2.7
Jan 27, 2025	Regular Expressions, Finite State Automata	1/28 Assignment 2 due 1/28 Assignment 3 (phone scraper)
Feb 3, 2025 (Remote classes this week)	Finite State Automata, Finite State Transducers	2/6 Assignment 3 due 2/6 Assignment 4 (regular expressions)
Feb 10, 2025	Introduction to Probability, Langauge Modeling	
Feb 17, 2025 (1 Session, Feb 20th)	Advanced Language Modeling	2/18 Assignment 4 due Jurafsky & Martin, Chapter 3.1 - 3.5
Feb 24, 2025	Midterm review **Thursday, February 27th: Midterm Exam (in class)**	2/25 Assignment 5 (language modeling) Jurafsky & Martin, Chapter 3.5 - 3.8
Mar 3, 2025	Spring Break! (No classes)	
Mar 10, 2025	Intro to Neural Network Language Models, Introduction to Markov Models	3/13 Assignment 5 due 3/13 Assignment 6 (part of speech tagging)
Mar 17, 2025	Sequence Tagging	3/20 Assignment 6 due 3/20 Assignment 7 (named entity recognition)
Mar 24, 2025	Named Entity Recognition, Introduction to Context-Free Grammars	
Mar 31, 2025	Context-Free Grammars and Parsing	4/1 Assignment 7 due 4/3 Assignment 8 (vector space models)
Apr 7, 2025	Vector Space Modeling	
Apr 14, 2025 (1 session, April 15th)	Naive Bayes Classifiers	4/15 Assignment 8 due
Apr 21, 2025	More Naive Bayes Classifiers, Wrap-up	4/24 Assignment 9 (practice final; not graded for correctness)

Apr 28, 2025 (1 session; April 29th)	Final Review	4/29 Assignment 8 due
May 2, 2025	**Final Exam: (12:30 - 2:30 pm)**	

Prerequisites: There are no prerequisites for this course. In particular, we will assume that students have no prior experience with programming. If you *do* have some limited prior experience with coding but not with NLP, this course is still likely appropriate for you. However, please speak with Ethan if you have any questions or concerns.

Grade Breakdown

Midterm Exam	25%
Final Exam	35%
Homework Assignments	30%
Participation and Attendance	10%

Participation and Attendance: This course is a challenging, fast-paced introduction to two different topics, computer programming and natural language processing. Given that we will be moving quickly between modules, it is very easy to fall behind. I encourage you to miss class only for things such as planned medical events, medical emergencies, family emergencies, or religious observations. While we will not be keeping track of attendance, formally, 10% of your grade will be assigned based on your participation. Participation can take many forms – asking questions in class, engaging in group activities, and attending office hours. For more information regarding the university policy on attendance, please see the academic standards section of the Undergraduate Student Bulletin (https://bulletin.georgetown.edu/regulations/standards/)

- Class time will involve personal and group coding sessions, so **please bring your laptop to class!**

Assignments: Over the semester, students will complete eight programming assignments and one practice final (graded for completion only). Assignments are due before the start of class on their specified due date. Grades for assignments are determined in the following way:

- **Original Submission:** The original submission will be graded for correctness. Due dates for the original submission are given in the above course schedule, however they may be adjusted throughout the semester.
    - **Late Policy:** Late homework can receive credit, but the grade you can achieve is capped based on how late it is. For the first two days, the maximum grade you can achieve is lowered by 5%, and after that by 10% per 24-hour period after the due time. If later than six days, you can receive up to 50% credit for the original submission.
- **Corrected Submission:** Separate from your original submission, you will be asked to submit a corrected version of the homework. In this corrected version, you should document the changes

necessary to turn your original submission into a corrected submission. Leave comments in the code noting each change you make to improve your original submission.

To facilitate corrections, we will go over homework solutions in class and provide in-class time for students to ask individual questions and help each other with corrections. We will accept corrections up to the final day of class, 4/29, however, I strongly encourage you to submit your correction shortly after we go over the solution in class.

- While permitted, if you received corrections help from someone (fellow student, dormate, sibling), please note who you asked for help somewhere in your resubmission.
- If your original homework submission was entirely correct, you can simply resubmit it as your "corrected" version.

All submissions happen through Canvas. There will be separate submission portals for the original submission and the corrected submission.

- **Group Work Policy:** The code that you submit as part of homework assignments *must be written by you*. However, working in groups on assignments can be extremely helpful. If you work with a group to come up with *pseudocode* (a high-level plan for how your program will run), or if you work with a group to understand and debug a piece of non-functioning code, that's OK. But you need to implement the pseudocode, or fix the bug yourself!

For more information on the University's honor code, please see https://honorcouncil.georgetown.edu/.

Readings: Most of the course readings will be derived from Jurafsky and Martin, 3rd edition, which can be accessed online here: https://web.stanford.edu/jurafsky/slp3/. Readings not from this book will be posted online in Canvas.

Midterm and Final: The midterm and final exams will not require live coding on a computer. Rather, they will involve commenting on or correcting code, explaining why and how code does (or doesn't) work, as well as explaining concepts and working with formulae we've covered in class. The TA will lead practice sessions in advance of each exam. The mid-term will take place during a class session.

Use of AI Assistants: The use of AI assistants, such as ChatGPT, Bard, or Claude, as well as coding plug-ins like Copilot, are not explicitly forbidden. These tools are increasingly part of our world and are based on the technologies we will learn about in this class. Indeed, one of the learning outcomes is to encourage critical thinking about the abilities and limitations of such systems.

With that in mind, a warning: While LLM-based tools can be extremely effective, they suffer from two major drawbacks. First, they often make mistakes. Second, they reduce the amount *you* have to think about and understand the code you run (that's the point!). If you are an advanced coder or an expert debugger, then you will be able to spot the mistakes that these systems make and fix them. However, for beginners, it's quite possible that correcting a buggy LLM-generated algorithm will take more time than writing the algorithm yourself. Furthermore, every time you rely on an LLM to generate something for

you, that's one less opportunity you have to solidify the concepts we learn in lectures and to practice coding as a skill. Importantly, you won't be able to use LLMs during the midterm or the final. If you use these systems extensively in the beginning you'll have a much harder time preparing for the exams.

Accommodations: If you have a recognized accommodation through ARC, please contact Ethan. More information about accommodations and support, including student-athlete support, can be found on the ARC website (https://academicsupport.georgetown.edu/)