LING/CSC/DATA/INFO/ISTA 439/539
Jackson
Spring 2026 (15-week)

Statistical Natural Language Processing

Course Overview

> This 3-credit course introduces the key concepts underlying statistical natural language processing. Students will learn a variety of techniques for the computational modeling of natural language, including: n-gram models, smoothing, Hidden Markov models, Bayesian inference, expectation maximization, the Viterbi algorithm, the Inside-Outside algorithm for probabilistic context-free grammars, and higher-order language models. *(from the course catalog)*

Course objectives

In this course, we will …

- cover fundamentals of natural language processing (NLP), such as setting up a development environment and performing text pre-processing (tokenization, normalization)
- apply text classification algorithms, such as **naive Bayes classifiers** and **logistic regression classifiers**, while learning basic principles of machine learning.
- explore a range of important NLP applications, such as options for **word representations** (contrasting one-hot vectors with embeddings), **sequence labeling** (part of speech tagging, shallow parsing/chunking, etc.), and **structured prediction** (chart-based parsing and/or transition-based dependency parsing)

This course prepares students with a solid understanding the foundations of the field so that they can understand state-of-the-art statistical NLP topics (as introduced in LING/CSC 582 *Advanced Statistical NLP*) more deeply.

Learning outcomes

After successfully completing this course, students will be able to…

- carry out a variety of natural language processing (NLP) tasks[1]
- compare techniques for word and document representations[1]
- implement a subset of the algorithms and architectures covered in this class[2]
- understand an NLP tool or approach well enough to explain it to others.[2]

[1] Relates to Linguistics Department's UG Program Outcome 1.
[2] Relates to Linguistics Department's HLT Program Outcomes 1, 2, & 3.

HLT program learning outcomes

By completion of the HLT program, students will be able to:

1. **Write, debug, and document readable and efficient code** in programming languages commonly used to develop, implement, and evaluate HLT models, as demonstrated through course projects and a professional internship.
2. **Select and apply appropriate algorithms and core concepts** in HLT to perform common tasks and solve realistic problems, as demonstrated through course projects and a professional internship.
3. **Apply common tools and libraries** used in HLT by integrating them into course projects and real-world applications or workflows, as demonstrated through course projects and a professional internship.
4. **Demonstrate professional skills** in the field of HLT, including effective teamwork, clear and concise communication, professional networking, understanding of business procedures and team-based code development, leadership, and critical thinking, as demonstrated through course presentations, projects, and a professional internship.

Prerequisites

- Programming competency in Python at the level of ISTA 130 or higher

Locations and times

This is an in-person course, but we will make use of online tools for certain learning activities.

Our in-person class sessions will be held on Mondays and Wednesdays (except for university holidays), 9:30am to 10:45am, in the Modern Languages building, room 205. Our first class session will be Wednesday, January 14th, and our last class scheduled session will be Wednesday, May 6th, for a total of 30 sessions.

Because I'm teaching both in-person courses and fully online courses this term, my office hours will be offered both in-person and online; regardless of your class modality, you may attend office hours in either format. Office hour times, location, and an online link can be found on the D2L instructor introduction page and here in the syllabus.

Please see the course D2L page for important dates and further information.

400/500 Co-convened Course Information

The difference between taking this course at the 500 level versus the 400 level is found in the **Assignments and grading** section below. Briefly, students in 539 will complete an open format, programming-based, work-at-home project which is not required for students in 439.

Instructor

name	Eric Jackson
email	ejackson1@arizona.edu
hours	Mondays 2:00pm–4:00pm (Arizona time, UTC-7) in person (COMM 114A) and online via Zoom at `https://arizona.zoom.us/j/85678652639` (passcode 816071), and by appointment.

My working hours are normal business hours in Arizona (M-F 9am-5pm), and I generally do my preparation and grading for the course at that time.

The best way to contact me is to send me an email or a forum message. I try to respond within 24 hours, but *during* working hours, I may be grading coursework, meeting with someone, or recording lectures, so my response may not come immediately. If there's a chance you may need my response, don't wait to discover that when a deadline is upon you.

I know that normal business hours in Arizona are not convenient for everyone, especially those in online asynchronous courses. If this describes you and you need to meet outside of Arizona business hours, I'm holding Thursday evenings open. Contact me in advance to set up a time and a link; I won't otherwise be online then.

Course components and content

Schedule / Progression of topics

Unit 0: *Your development environment.*
Introductory unit to get everyone working in a uniform development environment.

Unit 1: *NLP basics.*
A **brief** introduction to considerations when working with language as data.

Unit 2: *Machine learning basics,*
introduced in the context of naive Bayes classifiers.

Unit 3: *Logistic regression classifiers,*
which introduce many concepts that form the foundation of neural models.

Unit 4: *Word and phrase representations.*
How you represent units of language has a big effect on system performance.

Unit 5: *Sequence tagging.*
Use a Hidden Markov model find the proper part of speech of word tokens in a sentence

Unit 6: *Structured prediction.*
Use a shift-reduce parser to build a dependency parse for a sentence

Authoritative due dates for each unit and each assignment will be listed in D2L. Check the D2L course calendar to make sure you don't forget or miss a deadline.

Readings

A draft version of the textbook used in this course is available for free on-line.

> *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, Jurafsky and Martin, `https://web.stanford.edu/%7Ejurafsky/slp3/`

Additional readings may be available digitally in the course D2L site. In addition to the course textbook and any posted readings, students may find the following resources useful:

Bird, Steven, Ewan Klein, and Edward Loper. 2015. *Natural Language Processing with Python.* `https://www.nltk.org/book/`

Géron, Aurélien. 2019. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow.* (2nd ed.) O'Reilly. `http://neuralnetworksanddeeplearning.com/`

Goldberg, Yoav. 2017. *Neural Network Methods for Natural Language Processing.* Springer.

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning.* MIT press. `https://www.deeplearningbook.org/`

Nielsen, Michael. 2015. *Neural Networks and Deep Learning.* Springer. `http://neuralnetworksanddeeplearning.com/`

Attendance and participation

Students are expected to actively participate in the course by attending in-person class sessions, reading the assigned readings, completing the assigned learning activities and programming homework, and engaging with the instructor and other students in the course forum. I've assembled the class sessions and out-of-class activities to provide you with a solid learning experience. You're all adults, and you're responsible to invest time in your own learning.

If the content of a lecture is not clear, you are expected to ask a question in class, send a question to the instructor by email, meet with the instructor in regular office hours or arrange another time to meet, or post a question for clarification on the course forum.

The preferred place to ask questions about the course is on the on the course forum at `https://forum.hlt.arizona.edu/#narrow/channel/74-ling-539-inperson-sp2026`, not on D2L. If *you* have a question, it's possible that someone else has a similar question. Having the question and answer visible to the class on the forum means that everyone benefits from it. The course forum is also where you will also find course announcements.

If you have a question about a programming assignment that requires me to see your own code, that is not appropriate for a public forum post. You can send it in a direct message to me on the forum, or in an email.

For emergencies or for personal matters that you don't wish to put in a private post, please email the instructor.

Assignments and grading

All of the assignments have been designed to aid your learning and retention of course material. I expect everyone to attempt all of them, to gain the most from the course. The due date for each assignment will be posted with the assignment in D2L. All times will be given in Arizona time (Mountain Standard, GMT-7). Accepting late work would mean accepting work after I've already released detailed feedback on an assignment, or withholding timely feedback to the rest of the class on the issues in that assignment. **Except for university-approved reasons listed below, late work will not be accepted.**

There will be no in-class midterm or final exam. All graded items will be submitted online.

Review and Mastery activities (in TopHat via D2L) are an opportunity for you to practice applying the new concepts we learn in class, in a context where it's perfectly fine to not get things right. These online activities provide immediate feedback to you, to help you see where you might need to review class notes or readings, or ask for clarification.

Graded programming assignments will be given via GitHub Classroom, accessed by a link from the course website (D2L). These programming assignments should be completed in a uniform development environment using Python in a Jupyter Notebook and are graded using NBGrader. Test cases will be provided to indicate where your solutions have problems. ***Note: working on the assignment outside of Jupyter Notebook, for instance, in Google Colab, may introduce errors in the grading system such that your assignment is scored as a zero. I therefore recommend only using Jupyter.***

For students enrolled in 439, your overall course grade will be calculated based on this weighting of assignments:

type	number	total
programming assignments	4	70%
review and mastery / TopHat	11	30%

Additional work for students enrolled in 539:
A **private Kaggle competition** will provide an opportunity to apply the techniques learned in class to a set of real-world data. For this assignment, students will submit (a) a blog post describing your approach to the problem, (b) a GitHub repository with the source code for your solution, and (c) a submission to the class competition on Kaggle.com.

The overall course grade for students in 539 will be calculated based on this weighting of assignments:

type	number	total
programming assignments	4	65%
review and mastery	11	15%
class Kaggle competition	1	20%

A rubric for the class competition submission will be provided before the submission date, so that students know clearly what components their submission should include and how it will be graded.

Overall course grades will be calculated based on the following percentages:

Grade	Point Range
A	90 – 100
B	80 – 89
C	70 – 79
D	60 – 69
E	0 – 59

Technology

To complete your programming assignments, we recommend that you use a laptop or desktop with $\geq$ 8GB of RAM. All assignments and tutorials will be presented using a uniform Linux-based development environment which students will learn to configure during the zeroth unit of class. To complete your assignments, you will need ...

- A Linux desktop environment such as Ubuntu 24.04 LTS (possibly as a virtual machine)
- A GitHub account in order to submit through GitHub Classroom
- Docker (installed on your course-specific development environment)
- A modern web browser (such as Firefox or Chrome/Chromium)

Student collaboration, novel work, & appropriate AI use

The purpose of this course is to train **your** mind, and to do that, you need to **use** your own mind. You will gain the most benefit from the programming and other assignments in this course if **you** are the one who has come up with all the code, analysis, or examples, even if this requires a bit of mental struggle on your part to get it right. **Don't be afraid to struggle for a bit, and don't be afraid to submit work that may not be perfect, because the mental effort that you put into your own work is helping you learn.** Beyond a moderate amount of struggle, however, please seek outside help from the instructor.

Students are encouraged to discuss problems and general approaches for solutions with the instructor and with others in the course, but everyone must turn in **work that is the product of their own mind**. You may not submit assignments that are substantially the same as any other source (your classmates, someone online, or an AI tool). Using code from another source but simply changing the variable or object names is not sufficient to make it "your code".

If you do feel you need outside help, using portions of code you found online or created with Generative AI is acceptable, but it must constitute no more than 25% of your total code. If you obtain code other than writing it yourself, you must evaluate it critically **and clearly state on your assignment how much code is from other sources, also citing where it came from** (Stack Overflow, ChatGPT, CoPilot, etc).

If you discuss an assignment with a peer, if you find inspiration from a web resource, or if you use AI for appropriate help (ie, not simply copying and pasting its answer as your own), you must cite that fact on your assignment:

- "I discussed this assignment with Jane Studentname and Joe Wildcat."
- "I used ChatGPT for brainstorming of approaches to this coding task."
- "I wrote this code following a suggestion from StackOverflow at <URL>"

The general principle in all assignments is that the majority of the work you turn in must be new and must be your own. This has often caught international students by surprise; please pay special attention if this is different from the academic practices at your past institution. Do your own work, and please ask me in advance if you are unsure whether something will be acceptable or not. Assignments that do not seem to represent your own work—suspiciously similar to another student's or to a source online, or those that seem to have been mostly produced using generative AI—will be forwarded to the Dean of Students office in accordance with the Code of Academic Integrity (linked below). Please be a responsible adult and don't run the risk of losing credit for an assignment or for the course by copying, by allowing others to copy from you, or by having ChatGPT do your assignment for you.

I want to make certain that students understand the reasoning behind this policy:

Generative AI is a useful tool, like a calculator is a useful tool for doing math. However, in order to train students' cognitive skills, we ask them to learn to perform some math operations in their head before they default to using a calculator. The point of having students work through those problems without a calculator is not simply to get the right answer, but to develop their ability to think in certain ways. The same is absolutely true about the use of Generative AI.

In some contexts, being able to use a calculator is an important skill—while in other contexts, like when you're taking a math test to see whether you know basic math facts, solely using a calculator short-circuits your own learning. A bicycle is a tool that allows us to get from one place to another faster and more efficiently than running—but if you're going to be tested in your time for a 5k run, it won't help you to train for running solely by riding a bicycle. You will likely need to know how to use generative language models for tasks at some point, but having one write your homework or forum posts for this class is not appropriate. Put in the thinking yourself, so that you can reap the mental benefit for yourself.

Moreover, the metaphor of generative AI as a calculator is not perfect. Generative AI is like a calculator that sometimes returns untrustworthy results. You need to know how to perform these programming tasks on your own well enough that you can see where some AI-generated code is partially or completely off the mark, or introduces logic errors even if it runs without runtime errors.

The UA Library has a guide for students as to what is and is not appropriate use of AI and similar resources:

`https://libguides.library.arizona.edu/students-chatgpt/`

University boilerplate

All of the following items are required by the university to be included on syllabi. If you find something here that is surprising or unexpected, please bring it up with me as soon as possible.

By way of a brief summary:

Disabilities If you have a disability that affects how you will need to do the work in this class, please let me know *within the first week of class*.

Academic Code of Conduct Cheating and plagiarism are not remotely acceptable in any way. You are responsible for knowing whether your own behavior qualifies as plagiarism, and whether your use of AI is inappropriate. Disruptive behavior in class—which here includes audio, video, or text on any of our course websites or by email—is not acceptable. Please be respectful of others.

Sensitive Material This is a university and you are adults. It is possible that we may touch on topics that some students could find sensitive during the semester. Given the focus of this course, this seems unlikely, but I alert you nonetheless.

Health & Wellbeing

The university has a specific site for COVID information: `http://covid19.arizona.edu`. If you are experiencing personal or financial challenges from any health-related issue, let me know as soon as you can if we need to make accommodations, and please stay safe.

The semester ahead may come with ups and downs in both physical and mental health, but there are lots of ways to support yourself. Eat well, get regular exercise, and don't neglect things like self-care, talking with friends and family, or getting a fresh perspective from a supportive group. Stress is a normal part of life and may even motivate you sometimes, but chronic or overwhelming stress can affect your physical and mental health and wellbeing. Pay attention to your personal signs that you're overly stressed, like changes in your mood, appetite, sleep, behavior, or new physical symptoms (aches, pains, etc.) that interfere with school and daily life. If you notice these signs or have questions about helpful resources, I welcome you to talk with me. You can also visit `caps.arizona.edu/mental-health` for mental health tools and resources.

Mental Health & Wellness Resources

- **Health & Wellness:** Campus Health provides quality medical, mental health, and wellness services for students. Visit `health.arizona.edu` or call 520-621-9202 (520-570-7898 for help after hours)
- **Mental Health:** Campus Health's Counseling & Psych Services offers a range of mental health support tools and services like self-care strategies, peer support, groups and workshops, and professional mental health services. Visit `caps.arizona.edu/mental-health` or call CAPS 24/7 at 520-621-3334 to learn more.

- **Crisis Support:**

  Suicide & Crisis Lifeline: call 988 Crisis Text Line: text TALK to 741-741 Visit `preventsuicide.arizona.edu` for more suicide prevention tips and resources

Absence and Class Participation Policy

Attendance in an all-online course is not evaluated like attendance in an in-person course. For this course, attendance will be represented by active reading, completion, and participation in online course activities, including loading/viewing materials and completing activities posted on D2L, OpenClass, our course forum, and any other related websites.

The UA's policy concerning Class Attendance, Participation, and Administrative Drops is available at: `http://catalog.arizona.edu/policy/class-attendance-participation-and-administrative-drop`

The UA policy regarding absences is that any sincerely held religious belief, observance or practice will be accommodated where reasonable, `http://policy.arizona.edu/human-resources/religious-accommodation-policy`.

Absences pre-approved by the UA Dean of Students (or Dean Designee) will be honored. See: `https://deanofstudents.arizona.edu/absences`

Classroom Behavior Policy

To foster a positive learning environment, students and instructors have a shared responsibility. We want a safe, welcoming, and inclusive environment where all of us feel comfortable with each other and where we can challenge ourselves to succeed. To that end, our focus is on the tasks at hand and not on extraneous activities.

Students are asked to refrain from disruptive conversations with others in the course, including on asynchronous course platforms. Students observed engaging in disruptive activity will be asked to cease this behavior. Those who continue inappropriate behavior will be removed from that venue and may be reported to the Dean of Students.

Threatening Behavior Policy

The UA Threatening Behavior by Students Policy prohibits threats of physical harm to any member of the University community, including to oneself. See `http://policy.arizona.edu/education-and-student-affairs/threatening-behavior-students`.

Accessibility and Accommodations

At the University of Arizona, we strive to make learning experiences as accessible as possible. If you anticipate or experience barriers based on disability or pregnancy, please contact the Disability Resource Center (520-621-3268, `https://drc.arizona.edu/`) to establish reasonable accommodations.

Code of Academic Integrity

Students are encouraged to share intellectual views and discuss freely the principles and applications of course materials. However, graded work/exercises must be the product of independent effort unless otherwise instructed. **If you use a code snippet that you came up with from discussions with a classmate, that you found online, or even that you got from a large language model, it's important to cite where it came from, whether that source was Sally Classmate, GitHub.com, stackexchange.com, or ChatGPT.**

Students are expected to adhere to the UA Code of Academic Integrity as described in the UA General Catalog. See: `http://deanofstudents.arizona.edu/academic-integrity/students/academic-integrity`.

The UA Library provides a helpful learning module for students to understand and avoid plagiarism: `https://libguides.library.arizona.edu/info-strategies/plagiarism`

The UA Library also has resources to guide you to appropriate and safe use of AI and large language models: `https://libguides.library.arizona.edu/students-chatgpt/integrity`

UA Nondiscrimination and Anti-harassment Policy

The University is committed to creating and maintaining an environment free of discrimination; see

`http://policy.arizona.edu/human-resources/nondiscrimination-and-anti-harassment-policy`

Subject to Change Statement

Information contained in the course syllabus, other than the grade and absence policy, may be subject to change with advance notice, as deemed appropriate by the instructor.