Introduction to Natural Language Processing

LING-4400 (Fall 2023)
Mon+Wed 2-3:15

Instructor:
Amir Zeldes
E-Mail: amir.zeldes@georgetown.edu
Website: https://gucorpling.org/amir/
Office: Poulton 243 (office hours Wed 3:30-5:00)

Teaching Assistants:
Yilun Zhu, yz565@georgetown.edu
Jessica Lin, yl1290@georgetown.edu (office hours TBA)

Summary:
This course will introduce students to the basics of Natural Language Processing (NLP), a field which combines insights from Linguistics and Computer Science to produce applications such as machine translation, question answering, information retrieval, and more. We will cover a range of topics that will help students understand how current NLP technology works and will provide students with a platform for future study and research. We will learn to implement simple representations such as finite-state techniques, n-gram language models, part of speech tagging, topic modeling and basic parsing algorithms in the Python programming language, and familiarize ourselves with using pre-trained neural network libraries for these and other tasks. 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 understanding, along with an ability to assess the strengths and weaknesses of natural language technologies based on these methods.

Course requirements and final grade breakdown:

Homework assignments	30%
Mid-term	30%
Final exam	30%
Attendance & participation	10%

Approximate course plan

Week	Topics	Readings and activities	Assignments
Week 1	Introduction	Bar Hillel (1960), coreference exercise	Setup & why is NLP hard
Week 2	Python & NLTK basics		NLTK practice
Week 3	OOP Basics, CLI interface		Palindrome recognizer
Week 4	Tokenization & regular expressions	Jurafsky & Martin (2017), C2	
Week 5	Eliza chatbot		Extending Eliza
Week 6	Finite-state morphology	Jurafsky & Martin (2008), C2-3	Building morphological analyzers
Week 7		Morphology hackathon session	
Week 8	Language models, a shallow introduction to neural networks	Jurafsky & Martin (2017), C7	Story generator
Week 9		Neural LM	-- Midterm --
Week 10	Hidden Markov Models and sequence labeling	Sutton & McCallum (2006) (optional)	Viterbi-HMM POS tagger
Week 11	Parsing		Basic parser
		Collaborative grammar building	
Week 12	Topic models and LDA		TF-IDF for information retrieval
		Topics in the Reuters corpus	LDA
Week 13	Vector space models		Model training
Week 14			
	Conclusion		

We will discuss fundamentals of NLP such as word segmentation and part of speech tagging, syntactic parsing and computational morphology, as well as topics in document classification and topic modeling. In particular we will explore language models in detail, including statistical n-gram models a shallow introduction to neural language models, and their ability to predict/generate natural language input. We will learn about similarity metrics for strings and texts, and about search algorithms for efficient retrieval of the most likely solution within a set of possible ones. We will also learn how to construct and apply finite state models to the analysis of words in English and other languages, and how to use Hidden Markov Models and sequence labeling for the prediction of correct labels for sequences of words. We will apply dynamic programming using the Viterbi algorithm to find the optimal path through series of hidden states or labels, and how to use 'bag of words' models to separate large collections of documents into maximally distinct topics using Latent Dirichlet Allocation (LDA). All of these topics will be introduced step by step with accompanying Python starter code for each topic, which participants will have to modify and expand as the course progresses.

Participation, assignments, mid-term and final exam:
This course is a very intensive introduction, especially for those coming with less background in programming in general or Python in particular. This means that continuous attendance, submission of all homework assignments and attention to their correction is essential. In particular, homework assignments will involve including explanations for parts of each solution, and I will regularly call on students at random to present these to the class, so please make sure not only that your code works, but that you understand *why* it works. This is part of the homework: a good explanation of an imperfect submission can raise your score, but a bad explanation of working code can reduce it as well.

It is understood that, due to unforeseen circumstances (pandemic-related etc.), asynchronous participation may be required, but that students are responsible for reviewing class materials, following course progression, submitting assignments on time, and coming to class prepared. Students should alert the TAs or instructor if they get stuck, since skipping or not understanding earlier parts in the course can quickly translate into losing touch with the material.

Throughout the course we will refer to some reference works in natural language processing, which can also be helpful if catching up is required, including classics textbooks such as Jurafsky & Martin (2017) and Bird et al. (2017), with later optional readings from neural-network centric works (Goldberg 2017, Tunstall et al. 2022). However we will not work through these as textbooks per se, and relevant readings will be uploaded to Canvas. *Required readings* are considered a part of the assignments. Most assignments will include some aspects of coding, or expanding on existing code, but prose analyses of what our code is doing with some data, as well as other questions, will also be included. A useful reference for Python programming in general can be found in Lutz (2013).

The mid-term and final exam will not require live coding on a computer, and will not require students to remember complex code or formulas. Instead, the exam will concentrate on concepts we've learned in class, their application to language data, and analysis of given code which will have to be corrected, commented on or expanded in writing. All types of exam questions will be covered in examples as part of the homework assignments and practice questions will be provided ahead of time.

Use of AI plugins and conversational agents:
Using AI agents such as ChatGPT, Bing AI or Google Bard, as well as coding plugins such as GitHub Co-Pilot is not forbidden, unless explicitly specified (especially for some writing assignments). AI assistants are becoming a part of our world, and are based on NLP models themselves, so we should learn about them, and realistically, students will use them if they remain in the field.

However, when you are new to coding, especially at the beginning of the course, I do not recommend using AI agents or plugins *at all*. Using these effectively requires sufficient experience to understand when their suggestions are right, wrong, or something in between. At the level of this course, they are more likely to hamper learning than help: tackling a difficult problem head on by trying different things teaches you more than asking an agent to produce correct code. Relying on AI early on will probably make you less prepared for the mid-term and final, where no tools or reference materials are allowed. Later on, I recommend learning about and using AI tools as part of your toolkit, and we will discuss some of these later in the course.

Absences and timely assignment submission:
Students are expected to attend all classes and to complete all assignments on time. Absences may have an adverse effect on grades in a course, up to and including failure. That said, students may excuse themselves via e-mail from up to three meetings at their discretion, provided that they make up for lost course work and submit the assignments. Any additional absences for special reasons (religious observances, athletic travel, prolonged illness etc.) may be coordinated on a case by case basis with the instructor (documentation may be required as applicable). For this course in particular it is essential that any material missed by students is reviewed in depth until all points are understood – it is very easy to lose touch by missing some of the material, so please let me know about topics that remain unclear so we can discuss more in or out of class. Also make sure to make use of the TAs, who will offer office hours at different times beyond my own office hours – these are all resources meant to help you succeed, so please do not hesitate to make use of them.

I support a no-tolerance policy towards plagiarism, and would like to remind all students of their commitment to the Georgetown Honor System. While it is fine to ask others for help in order to understand topics in the material or in programming in general, homework assignments are your own to accomplish. In case of suspicious submissions I reserve the right to request clarification from relevant parties, including ensuring that authors of an assignment can explain the details of their submission orally. Assignments should be submitted via Canvas or if specified by e-mail, in which case they should include LING-4400 in the subject line. Late submission is subject to demerit points.

References

Bird, S., Klein, E., & Loper, E. (2017). *Natural Language Processing with Python*. Sebastopol, CA: O'Reilly.

Downey (2015) *Think Python*. Sebastopol, CA: O'Reilly.

Jurafsky, D., & Martin, J. H. (2017). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. 2nd edition. Upper Saddle River, NJ: Prentice Hall.

Lutz, M. (2013) *Learning Python*. Sebastopol, CA: O'Reilly.
Manning, C., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. Cambridge, MA: MIT Press.
Tunstall et al. (2022). *Natural Language Processing with Transformers*. Sebastopol, CA: O'Reilly.

Notice regarding sexual misconduct:
Please know that as a faculty member I am committed to supporting survivors of sexual misconduct, including relationship violence, sexual harassment and sexual assault. University policy also requires me to report any disclosures about sexual misconduct to the Title IX Coordinator, whose role is to coordinate the University's response to sexual misconduct.

Georgetown has a number of fully confidential professional resources who can provide support and assistance to survivors of sexual assault and other forms of sexual misconduct. These resources include:

Jen Schweer, MA, LPC
Associate Director of Health Education Services for Sexual Assault Response and Prevention
(202) 687-0323
jls242@georgetown.edu

Erica Shirley, Trauma Specialist
Counseling and Psychiatric Services (CAPS)
(202) 687-6985
els54@georgetown.edu

More information about campus resources and reporting sexual misconduct can be found at http://sexualassault.georgetown.edu.

Please fill out and return to me:

Name:
Degree/Major etc.:

What are your main learning goals for this course? What would you like to know or be able to do after taking it?

What operating system do you use?
[ ] Linux : _____ [ ] Mac [ ] Windows [ ] Other: _____

Do you have any programming experience and if so in what language?

Have you taken an introductory stats class?

Other comments: