CS 4650: Natural Language Processing

Wei Xu

Administrivia

- Course website:
  https://cocoxu.github.io/CS4650_fall2026/
  - slides, readings
  - course policies

- Ed Discussion:
  - for all class announcements, homework discussion, and contacting teaching staff
  - TA will start a mega-thread when release each assignment, and post a sign-up list for OH, etc

- Gradescope:
  - for homework submission and grading

Instructor

Wei Xu

Office Hours: Monday after class

Teaching Assistants

Brandon Michaels

Jameel Maayah

Minju Gwak

Rahul Vodeb Ghosh

Sharada Ghantasala

Yiren Wang

NLP X Research Lab

Director

Wei Xu

Associate Professor

Georgia Tech Machine Learning

LLM Training and Agents

- Post-training and reinforcement learning
- Reasoning and long-context LLMs
- Interactive and agentic AI
- AI safety and privacy


Tarek Naous


Geyang Guo


Jonathan Zheng


Duong Minh Le


Junmo Kang


Ivy He


Usneek Singh


Elene Sturua


Jiangyue Zhu

LLM Evaluation, Languages, and Culture

- LLM evaluation and benchmarks
- Multilingual LLMs
- Cultural adaptation and personalization


Yiren Wang


Frank Yang


Baixi Jiao


Benjamin Mamut


Victor Gong


Isha Madanu


James Balak


Sean Connelly


Alexey Plagov

Interdisciplinary Research

- AI for healthcare and medicine
- AI for science
- AI assistants for reading and writing

NSF

IARPA

DARPA

NIH

Google

SONY

criteo

I am Wei, an associate professor in CS, leading the NLP X lab. I and many of my students work on various aspects of large language models (LLMs) and vision-language models (VLMs) – I will show you a few examples next – and primarily publish at top NLP and ML conferences.

NLP X Research Lab

Director

Wei Xu

Associate Professor

Georgia Tech Machine Learning

Yao Dou, Benjamin Mamut, Wei Xu.

"Gavel: Agent Meets Checklist for Evaluating LLMs on Long-Context Legal Summarization" **(EMNLP 2026)**

Paper on arXiv

SimulatorArena: Are User Simulators Reliable Proxies for Multi-Turn Evaluation of AI Assistants?

Yao Dou[1] Michel Galley[2] Baolin Peng[2] Chris Kedzie[2†] Weixin Cai[2]
Alan Ritter[1] Chris Quirk[2] Wei Xu[1] Jianfeng Gao[2]
[1]Georgia Institute of Technology [2]Microsoft

douy@gatech.edu, mgalley@microsoft.com aka.ms/SimulatorArena

Abstract

Large language models (LLMs) are increasingly used in interactive applications, and human evaluation remains the gold standard for assessing their performance in multi-turn conversations. Since human studies are costly, time-consuming, and hard to reproduce, recent work explores using LLMs to simulate users for automatic assistant evaluation. However, there is no benchmark or systematic study to evaluate whether these simulated users are reliable stand-ins for real users. To address this, we introduce SimulatorArena, a benchmark of 909 annotated human–LLM conversations on two interactive tasks—math tutoring and document creation. SimulatorArena evaluates simulators based on how closely their messages match human behavior and how well their assistant ratings align with human judgments. Experiments on various simulator methods show that simulators conditioned on user profiles, capturing traits like background and message styles, align closely with human judgments. They reach Spearman's ρ of 0.7 on both tasks, providing a practical, scalable alternative to human evaluation. Using the best simulator for each task, we benchmark 18 assistants, including the latest LLMs such as GPT-5, Claude 4.1 Opus, and Gemini 2.5 Pro.

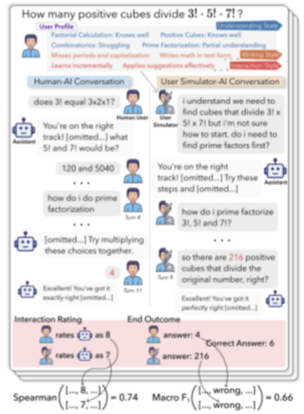


Figure 1: SimulatorArena systematically evaluates user simulators by comparing their behavior to humans'. User profiles improve simulator quality, offering an efficient, scalable alternative to human evaluation.

NeurIPS 2026 Workshop on Evaluation of Interactive Agents

https://eval-interactive-agents-workshop.github.io/

In a new paper just accepted into EMNLP 2026 main conference, we investigate LLMs' ability to handle long contexts, with a particular focus on multi-document summarization in the legal and medical domains. We plan to further expand this research direction in future work.

NLP X Research Lab

Director

Wei Xu

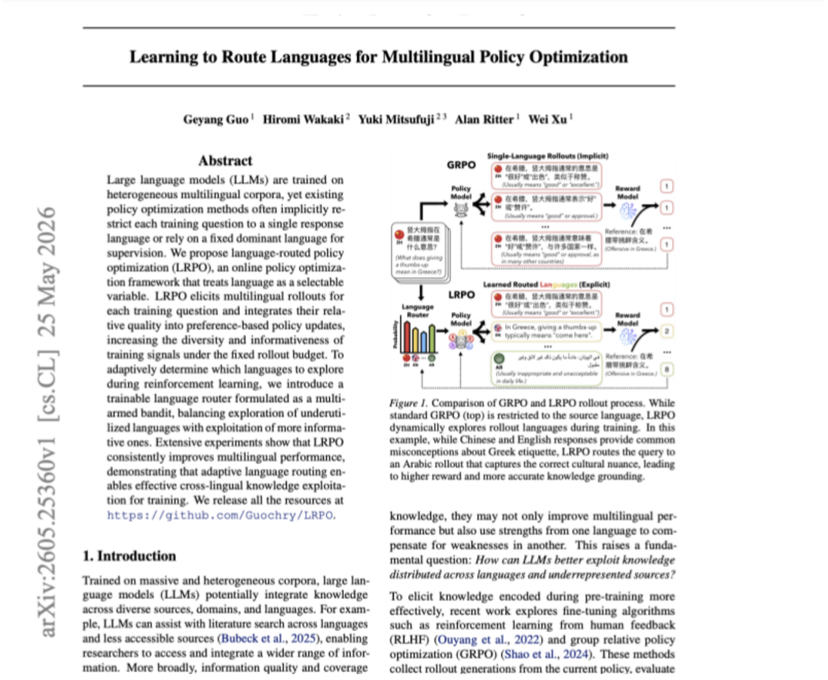
Associate Professor

Georgia Tech Machine Learning

Geyang Guo, Hiromi Wakaki, Yuki Mitsufuji, Alan Ritter, Wei Xu.

***"Learning to Route Languages for Multilingual Policy Optimization"* (ICML 2026)**

Paper on arXiv



Learning to Route Languages for Multilingual Policy Optimization

Geyang Guo [1] Hiromi Wakaki [2] Yuki Mitsufuji [2 3] Alan Ritter [1] Wei Xu [1]

Abstract

Large language models (LLMs) are trained on heterogeneous multilingual corpora, yet existing policy optimization methods often implicitly restrict each training question to a single response language or rely on a fixed dominant language for supervision. We propose language-routed policy optimization (LRPO), an online policy optimization framework that treats language as a selectable variable. LRPO elicits multilingual rollouts for each training question and integrates their relative quality into preference-based policy updates, increasing the diversity and informativeness of training signals under the fixed rollout budget. To adaptively determine which languages to explore during reinforcement learning, we introduce a trainable language router formulated as a multi-armed bandit, balancing exploration of underutilized languages with exploitation of more informative ones. Extensive experiments show that LRPO consistently improves multilingual performance, demonstrating that adaptive language routing enables effective cross-lingual knowledge exploitation for training. We release all the resources at https://github.com/Guochry/LRPO.

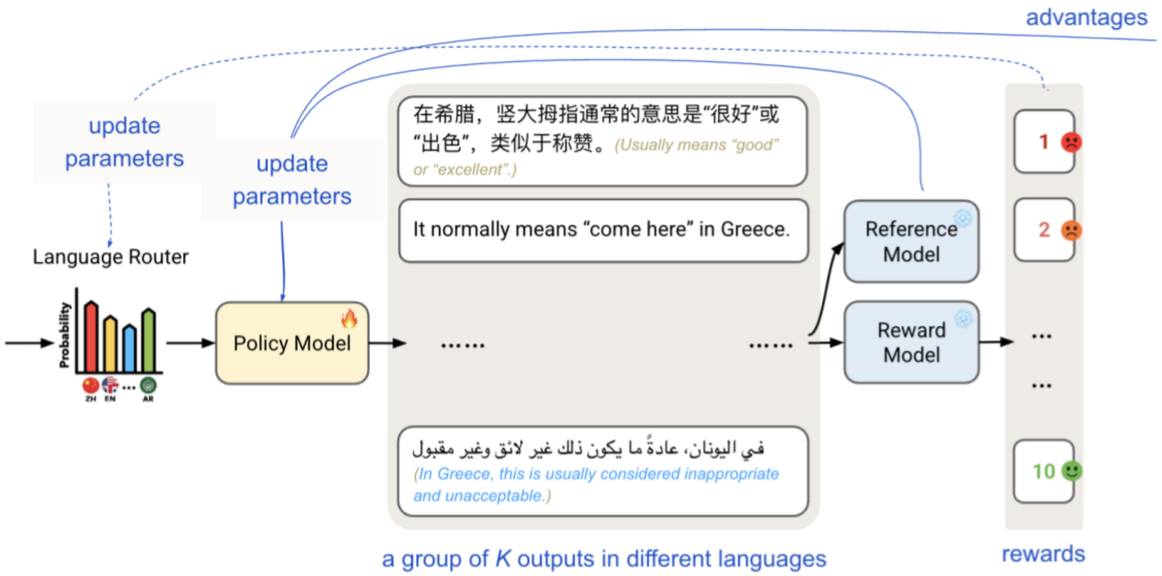
1. Introduction

Trained on massive and heterogeneous corpora, large language models (LLMs) potentially integrate knowledge across diverse sources, domains, and languages. For example, LLMs can assist with literature search across languages and less accessible sources (Bubeck et al., 2025), enabling researchers to access and integrate a wider range of information. More broadly, information quality and coverage

Figure 1. Comparison of GRPO and LRPO rollout process. While standard GRPO (top) is restricted to the source language, LRPO dynamically explores rollout languages during training. In this example, while Chinese and English responses provide common misconceptions about Greek etiquette, LRPO routes the query to an Arabic rollout that captures the correct cultural nuance, leading to higher reward and more accurate knowledge grounding.

knowledge, they may not only improve multilingual performance but also use strengths from one language to compensate for weaknesses in another. This raises a fundamental question: *How can LLMs better exploit knowledge distributed across languages and underrepresented sources?*

To elicit knowledge encoded during pre-training more effectively, recent work explores fine-tuning algorithms such as reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022) and group relative policy optimization (GRPO) (Shao et al., 2024). These methods collect rollout generations from the current policy, evaluate



In a paper published at ICML 2026, we worked on multilingual reinforcement learning for post-training of LLMs. By framing language selection as a multi-armed bandit problem, LRPO dynamically balances exploration and exploitation across different languages, improving the diversity and effectiveness of training signals during reinforcement learning.

NLP X Research Lab

Director

Wei Xu

Associate Professor

Georgia Tech Machine Learning

Jonathan Zheng, Sauvik Das, Alan Ritter, Wei Xu. ***"Probabilistic Reasoning with LLMs for Privacy Risk Estimation"*** **(NeurIPS 2025)**

Paper on arXiv

Probabilistic Reasoning with LLMs for Privacy Risk Estimation

Jonathan Zheng[1] **Sauvik Das**[2] **Alan Ritter**[1] **Wei Xu**[1]
[1]School of Interactive Computing, Georgia Institute of Technology
[2]Human-Computer Interaction Institute, Carnegie Mellon University
jzheng324@gatech.edu

Abstract

Probabilistic reasoning is a key aspect of both human and artificial intelligence that allows for handling uncertainty and ambiguity in decision-making. In this paper, we introduce a new numerical reasoning task under uncertainty for large language models, focusing on estimating the privacy risk of user-generated documents containing privacy-sensitive information. We propose BRANCH, a new LLM methodology that estimates the k-privacy value of a text—the size of the population matching the given information. BRANCH factorizes a joint probability distribution of personal information as random variables. The probability of each factor in a population is estimated separately using a Bayesian network and combined to compute the final k-value. Our experiments show that this method successfully estimates the k-value 73% of the time, a 13% increase compared to o3-mini with chain-of-thought reasoning. We also find that LLM uncertainty is a good indicator for accuracy, as high variance predictions are 37.47% less accurate on average.

1 Introduction

Large language models (LLMs) have shown increasingly strong performance in mathematical and logical reasoning [84, 61, 16], enabling exploration of a broad suite of user-facing applications. One such application is helping users understand the magnitude of privacy risks when disclosing personal information online — a holy grail of usable privacy that remains elusive [59, 38]. For example, how much riskier is it to disclose one's precise age versus a general age category (22 vs. 18-25)? To that end, we introduce a task that instructs LLMs to estimate the k number of people in the world that match the personal attributes or experiences presented in user-written messages on pseudonymous online fora like Reddit [18] or anonymous user interactions with ChatGPT [88, 50]. This task serves both as a new challenge for LLMs' reasoning abilities beyond conventional math and logic tests [28, 45, 34] and as a practical security tool to inform users about online safety.

Chain of Thoughts: **Instruction:** How many people share all the following attributes as this user?

Conditional Dependency Errors (25.64%)

The poster is from **Italy**, **female**, **26 years old**, **on the spectrum**, and has **social anxiety**.

1. Gender: The population of Italy is 59,000,000. ~50% are female. 29,500,000 × 0.5 = 29,500,000
2. Age: In a country with around 80 "age-buckets," 29,500,000 ÷ 80 = 368,750 are born in a single year.
3. Autism: General prevalence for women is around 0.25%. 368,750 × 0.0025 ≈ 921.88
4. Social anxiety: Estimates for people with social anxiety are around 10%. 921.88 × 0.1 ≈ 92.19
   (The prevalence of social anxiety in people on the Autism spectrum is roughly 50%)

After applying all these "filters" sequentially we get an estimate of roughly **93** people share these traits. (Correct: 461)

Personally Identifiable Information Errors (15.38%)

The poster is an **entrepreneur** of a **software development services company** called **Townbridge InfoTech**.

1. Occupation: There are 500 million native English speakers. 5% (25 million) of people are entrepreneurs.
2. Business: Roughly 10% entrepreneurs are launching IT companies. 25,000,000 × 0.1 = 2,500,000.
3. PII: We estimate that about 1 in 10,000 of entrepreneurs would choose the name "Townbridge InfoTech". 2,500,000 × 0.0001 = 250 (Business names as specific as "Townsbridge InfoTech" are unique, i.e. 1 in 2,500,000)

Based on these self-disclosures, approximately **250** people would share all of these specific details. (Correct: 1)

Figure 1: Most common Chain-of-Thought reasoning error types (with occurrence rates) and examples for o3-mini on Privacy Risk Estimation. Errors and correct explanations are highlighted. Chain-of-Thought struggles to model PII and capture relationships between attributes for risk assessments.

Methods	Models	Spearman's ρ ↑	Log error ↓	Within Range ↑
Chain of Thoughts	GPT-4o (2024-08-06)	0.654	3.04	56.29%
	DeepSeek R1 (2025-01-20)	0.693	2.93	56.95%
	o3-mini (2025-01-31)	0.729	2.39	64.90%
BRANCH (our work)	GPT-4o (2024-08-06)	0.797	2.16	66.89%
	o3-mini (2025-01-31)	0.817	2.04	72.19%
Human	–––––	0.916	1.57	78.79%

In another paper published at NeurIPS 2025, we focus on probabilistic reasoning methods with LLMs for estimating privacy risk in documents that contains user information. By reasoning via an Bayesian network framework, our approach BRANCH first factorizes into sub-questions before reconstructing the joint distribution, outperforming the Chain-of-Thought prompting.

NLP X Research Lab

Director

Wei Xu

Associate Professor

Georgia Tech Machine Learning

Tarek Naous, Michael J. Ryan, Alan Ritter, Wei Xu. *"Having Beer After Prayer? Measuring Cultural Bias in LLMs"*
(ACL 2024 - 🏆 Best Social Impact Award)

Paper on arXiv

Having Beer after Prayer?
Measuring Cultural Bias in Large Language Models

Tarek Naous, Michael J. Ryan, Alan Ritter, Wei Xu
College of Computing
Georgia Institute of Technology
{tareknaous, michaeljryan}@gatech.edu; {alan.ritter, wei.xu}@cc.gatech.edu

Abstract

As the reach of large language models (LMs) expands globally, their ability to cater to diverse cultural contexts becomes crucial. Despite advancements in multilingual capabilities, models are not designed with appropriate cultural nuances. In this paper, we show that multilingual and Arabic monolingual LMs exhibit bias towards entities associated with Western culture. We introduce CAMEL, a novel resource of 628 naturally-occurring prompts and 20,368 entities spanning eight types that contrast Arab and Western cultures. CAMEL provides a foundation for measuring cultural biases in LMs through both extrinsic and intrinsic evaluations. Using CAMEL, we examine the cross-cultural performance in Arabic of 16 different LMs on tasks such as story generation, NER, and sentiment analysis, where we find concerning cases of stereotyping and cultural unfairness. We further test their text-infilling performance, revealing the incapability of appropriate adaptation to Arab cultural contexts. Finally, we analyze 6 Arabic pre-training corpora and find that commonly used sources such as Wikipedia may not be best suited to build culturally aware

Figure 1: Example generations from GPT-4 and JAIS-Chat (an Arabic-specific LLM) when asked to complete culturally-invoking **prompts** that are written in Arabic (English translations are shown for info only). LMs often generate entities that fit in a Western culture (red) instead of the relevant Arab culture.

Press Coverage

Credit: VentureBeat made with Midjourney

We are also interested in multilingual and multicultural aspects of LLMs. One of our work led by my PhD student Tarek Naous, with undergrad Michael Ryan, looked into the cultural aspects in multilingual large language models. As LLMs are being deployed more widely around the world, it is very important for the LLMs to adapt to different cultures. It got press coverage by VentureBeat.

NLP X Research Lab

Director

Wei Xu
Associate Professor

Georgia Tech Machine Learning

"Stanceosaurus: Classifying Stance Towards Multicultural Misinformation"
Jonathan Zheng, Ashutosh Baheti, Tarek Naous, Wei Xu, Alan Ritter (EMNLP 2022)

"Revisiting non-English Text Simplification: A Unified Multilingual Benchmark"
Michael J. Ryan, Tarek Naous, Wei Xu (ACL 2023)

"Having Beer After Prayer? Measuring Cultural Bias in LLMs"
Tarek Naous, *Michael J. Ryan*, Alan Ritter, Wei Xu (ACL 2024 - 🏆 Best Social Impact Award)

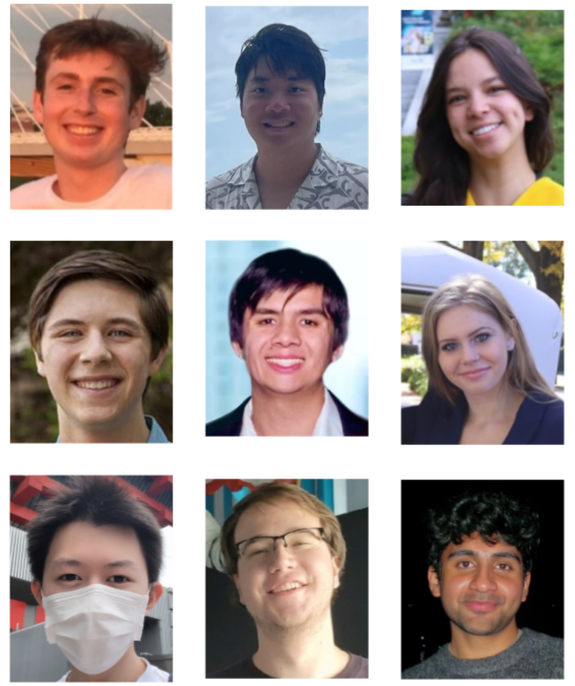
"Improving Minimum Bayes Risk Decoding with Multi-Prompt"
David Heineman, Yao Dou, Wei Xu (EMNLP 2024)

"Dancing Between Success and Failure: Edit-level Simplification Evaluation using SALSA"
David Heineman, Yao Dou, Wei Xu (EMNLP 2024)

"*What are Foundation Models Cooking in the Post-Soviet World?*"
Anton Lavrouk, Tarek Naous, Alan Ritter, Wei Xu (EMNLP 2024)

"The Impact of Visual Information in Chinese Characters: Evaluating Large Models' Ability to Recognize and Utilize Radicals"
Xiaofeng Wu, Karl Stratos, Wei Xu (ACL 2025)

"*Comparing Media Framing of AI Across Global Powers*"
Julie O Young*, Katerina Addington*, Yiren Wang*, Geyang Guo*, others, Alan Ritter, Wei Xu (in submission 2026)



I have had the privilege of supervising several Georgia Tech undergraduates on their undergraduate theses or MS theses (for BS/MS students). One of them, David Heineman, received the Georgia Tech College of Computing Outstanding Undergraduate Award 🏅 in 2024—awarded to just one student among 3,000+ CS undergraduates each year.

Course Goals

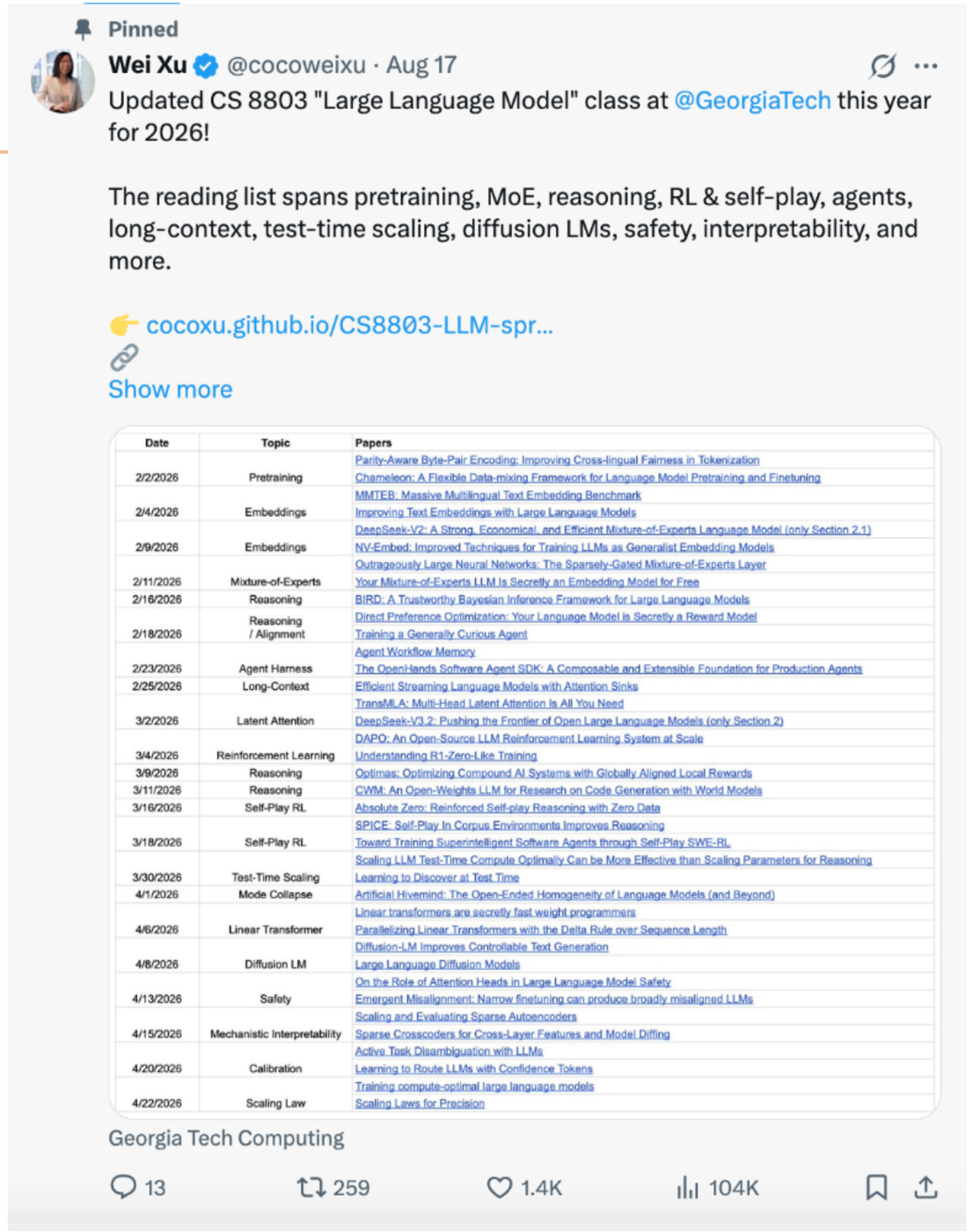
- Cover fundamental machine learning techniques used in NLP
- Understand how to look at language data and approach linguistic phenomena
- Cover modern NLP problems encountered in the literature:
- Make you a “producer” rather than a “consumer” of NLP tools
  - The four programming assignments will teach you the core concepts needed to understand the deep learning models used in nearly any NLP research paper or system.

Scope of the Class

- Consider this class as “Deep Learning for Natural Language Processing”
- Take this class only after you’ve taken CS 4641. We only provide a quick recap of basic ML models to deepen your understanding.
- It will cover the major deep learning architectures, including feedforward, recurrent, convolutional, and Transformer models. (similar to CS 4644/7643)
- It will cover classic NLP problems, such as part-of-speech, sequential tagging, machine translation, etc.
- It will also cover some basics of language models, especially how it started, why people design LLMs the way it is today, etc.

Scope of the Class

- However, CS 4650 will not cover large language models in-depth.
- For that, we recently developed a new course, CS 7652: Large Language Models.
- CS 4650 provides a good foundation before taking CS 7652.



Pinned

Wei Xu @cocoweixu · Aug 17

Updated CS 8803 "Large Language Model" class at @GeorgiaTech this year for 2026!

The reading list spans pretraining, MoE, reasoning, RL & self-play, agents, long-context, test-time scaling, diffusion LMs, safety, interpretability, and more.

👉 cocoxu.github.io/CS8803-LLM-spr...

Show more

Date	Topic	Papers
2/2/2026	Pretraining	Parity-Aware Byte-Pair Encoding: Improving Cross-lingual Fairness in Tokenization Chameleon: A Flexible Data-mixing Framework for Language Model Pretraining and Finetuning
2/4/2026	Embeddings	MMTEB: Massive Multilingual Text Embedding Benchmark Improving Text Embeddings with Large Language Models
2/9/2026	Embeddings	DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model (only Section 2.1) NV-Embed: Improved Techniques for Training LLMs as Generalist Embedding Models
2/11/2026	Mixture-of-Experts	Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer Your Mixture-of-Experts LLM Is Secretly an Embedding Model for Free
2/16/2026	Reasoning	BIRD: A Trustworthy Bayesian Inference Framework for Large Language Models
2/18/2026	Reasoning / Alignment	Direct Preference Optimization: Your Language Model is Secretly a Reward Model Training a Generally Curious Agent
2/23/2026	Agent Harness	Agent Workflow Memory The OpenHands Software Agent SDK: A Composable and Extensible Foundation for Production Agents
2/25/2026	Long-Context	Efficient Streaming Language Models with Attention Sinks
3/2/2026	Latent Attention	TransMLA: Multi-Head Latent Attention Is All You Need DeepSeek-V3.2: Pushing the Frontier of Open Large Language Models (only Section 2)
3/4/2026	Reinforcement Learning	DAPO: An Open-Source LLM Reinforcement Learning System at Scale Understanding R1-Zero-Like Training
3/9/2026	Reasoning	Optimas: Optimizing Compound AI Systems with Globally Aligned Local Rewards
3/11/2026	Reasoning	CWM: An Open-Weights LLM for Research on Code Generation with World Models
3/16/2026	Self-Play RL	Absolute Zero: Reinforced Self-play Reasoning with Zero Data
3/18/2026	Self-Play RL	SPICE: Self-Play In Corpus Environments Improves Reasoning Toward Training Superintelligent Software Agents through Self-Play SWE-RL
3/30/2026	Test-Time Scaling	Scaling LLM Test-Time Compute Optimally Can be More Effective than Scaling Parameters for Reasoning Learning to Discover at Test Time
4/1/2026	Mode Collapse	Artificial Hivemind: The Open-Ended Homogeneity of Language Models (and Beyond)
4/6/2026	Linear Transformer	Linear transformers are secretly fast weight programmers Parallelizing Linear Transformers with the Delta Rule over Sequence Length
4/8/2026	Diffusion LM	Diffusion-LM Improves Controllable Text Generation Large Language Diffusion Models
4/13/2026	Safety	On the Role of Attention Heads in Large Language Model Safety Emergent Misalignment: Narrow finetuning can produce broadly misaligned LLMs
4/15/2026	Mechanistic Interpretability	Scaling and Evaluating Sparse Autoencoders Sparse Crosscoders for Cross-Layer Features and Model Diffing
4/20/2026	Calibration	Active Task Disambiguation with LLMs Learning to Route LLMs with Confidence Tokens
4/22/2026	Scaling Law	Training compute-optimal large language models Scaling Laws for Precision

Georgia Tech Computing

13 259 1.4K 104K

Outline of the Course

ML and structured prediction for NLP

Deep Learning (Neural Networks)

Language Models

	Topic	*Projects*	*Problem Sets*
8/24/2026	Course Overview	Proj. 0 Out	PS0 Out
8/26/2026	Guest Lecture		PS0 Due (8/27), PS1 Out
8/31/2026	Machine Learning Recap - Naive Bayes, MLE logistic regression, perceptron,	PyTorch Tutorial	
9/2/2026	Machine Learning Recap - SVM, multi-class classification	Proj. 0 Due (9/3), Proj. 1 Out	
9/7/2026	**No class - Labor Day holiday** 💪		
9/9/2026	Neural Networks - feedforward network, training, optimization		
9/14/2026	Word Embeddings		PS1 Due (9/17)
9/16/2026	Sequence Labeling		
9/21/2026	Conditional Random Fields	Proj. 1 Due (9/24), Proj. 2 Out	
9/23/2026	Recurrent Neural Networks, Neural CRF		PS2 Out
9/28/2026	Encoder-Decoder, Attention	Instructions Out for a (lightweight) project proposal	
9/30/2026	Transformer		
10/5/2026	**No class - Fall Break** 🍂		
10/7/2026	MT Evaluation, CNN		
10/12/2026	Pretrained Language Models (part 1 - ELMo, BERT, BART/T5, Variants of BERT models)	Proj. 2 Due (10/13)	
10/14/2026	Pretrained Language Models (part 2 - GPT2/3, knowledge distillation, decoding)		PS2 Due (10/20)
10/19/2026	Pretrained Language Models (part 3 - Post-training of Language Models), midterm review		
10/21/2026	Pretrained Language Models (part 4 - Open-source Language Models)		
10/26/2026	**potential date for Midterm** 📝		
10/28/2026	**potential date for Midterm** 📝 (midterm date may change if there is unexpected event)		10/31 withdraw deadline
11/2/2026	Pretrained Language Models (part 5 - Tokenization, Multilingual NLP)	Proj. 3 Out	
11/4/2026	Pretrained Language Models (part 6)		
11/9/2026	**student in-class presentation**	Slides for In-class Presentation Due (11/9)	
11/11/2026	**Guest Lecture**	Course Project Proposal Due (11/17)	
11/16/2026	**student in-class presentation**		
11/18/2026	**student in-class presentation**		
11/23/2026	**Guest Lecture**	Proj. 3 Due (11/23) - last homework to use flexible day	
11/25/2026	No class - school recess 🦃		
11/30/2026	**student in-class presentation**		
12/2/2026	**student in-class presentation**		
12/7/2026	last instruction day		

tentative plan

(subject to change)

* Link to this Google spreadsheet on course website:
https://docs.google.com/spreadsheets/d/1oQO5pFEWfOYj0n64h4IYo-EwjWFrEql53dck1TA8Jdo/edit?usp=sharing

Course Requirements

- **Probability** (e.g. conditional probabilities, conditional independence, Bayes Rule)
- **Linear Algebra** (e.g., multiplying vectors and matrices, matrix inversion)
- **Multivariable Calculus** (e.g., calculating gradients of functions with several variables)
- **Programming / Python experience** (medium-to-large scale project, **debug** PyTorch codes when there are no error messages)
- Prior exposure to machine learning

There will be a lot of math and programming!

Some Example Slides

Sequential Models - e.g., Conditional Random Fields

- Model: $P(\mathbf{y}|\mathbf{x}) = \frac{1}{Z}\prod_{i=2}^{n}\exp(\phi_t(y_{i-1}, y_i))\prod_{i=1}^{n}\exp(\phi_e(y_i, i, \mathbf{x}))$

$$P(\mathbf{y}|\mathbf{x}) \propto \exp w^\top \left[\sum_{i=2}^{n} f_t(y_{i-1}, y_i) + \sum_{i=1}^{n} f_e(y_i, i, \mathbf{x})\right]$$

- Inference: argmax P(**y**|**x**) from Viterbi

- Learning: run forward-backward to compute posterior probabilities; then

$$\frac{\partial}{\partial w}\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=1}^{n} f_e(y_i^*, i, \mathbf{x}) - \sum_{i=1}^{n}\sum_{s} P(y_i = s|\mathbf{x}) f_e(s, i, \mathbf{x})$$

Some Example Slides

Training CRFs

$$\frac{\partial}{\partial w}\mathcal{L}(\mathbf{y}^*, \mathbf{x}) = \sum_{i=2}^{n} f_t(y^*_{i-1}, y^*_i) + \sum_{i=1}^{n} f_e(y^*_i, i, \mathbf{x}) - \mathbb{E}_{\mathbf{y}}\left[\sum_{i=2}^{n} f_t(y_{i-1}, y_i) + \sum_{i=1}^{n} f_e(y_i, i, \mathbf{x})\right]$$

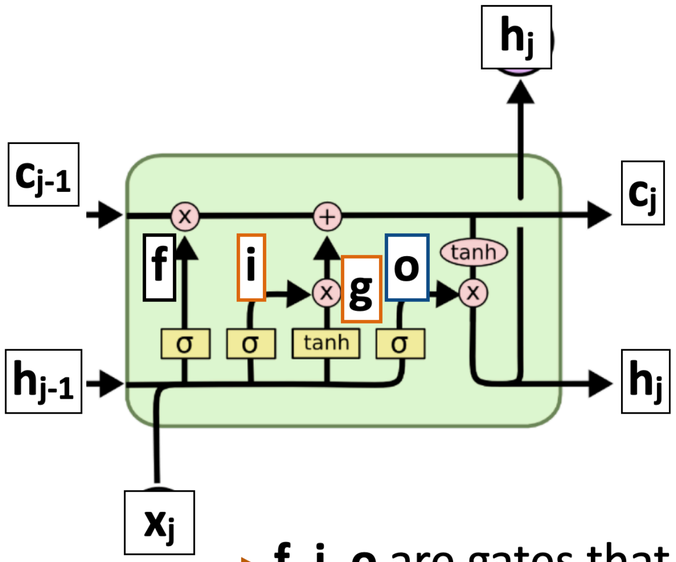
- Let's focus on emission feature expectation

$$\mathbb{E}_{\mathbf{y}}\left[\sum_{i=1}^{n} f_e(y_i, i, \mathbf{x})\right] = \sum_{\mathbf{y}\in\mathcal{Y}} P(\mathbf{y}|\mathbf{x})\left[\sum_{i=1}^{n} f_e(y_i, i, \mathbf{x})\right] = \sum_{i=1}^{n}\sum_{\mathbf{y}\in\mathcal{Y}} P(\mathbf{y}|\mathbf{x}) f_e(y_i, i, \mathbf{x})$$

$$= \sum_{i=1}^{n}\sum_{s} P(y_i = s|\mathbf{x}) f_e(s, i, \mathbf{x})$$

Some Example Slides

Neural Network Models — e.g., LSTMs



$$\mathbf{c_j} = \mathbf{c_{j-1}} \odot \mathbf{f} + \mathbf{g} \odot \mathbf{i}$$

$$\mathbf{f} = \sigma(\mathbf{x_j W^{xf}} + \mathbf{h_{j-1} W^{hf}})$$

$$\mathbf{g} = \tanh(\mathbf{x_j W^{xg}} + \mathbf{h_{j-1} W^{hg}})$$

$$\mathbf{i} = \sigma(\mathbf{x_j W^{xi}} + \mathbf{h_{j-1} W^{hi}})$$

$$\mathbf{h_j} = \tanh(\mathbf{c_j}) \odot \mathbf{o}$$

$$\mathbf{o} = \sigma(\mathbf{x_j W^{xo}} + \mathbf{h_{j-1} W^{ho}})$$

- **f**, **i**, **o** are gates that control information flow
- **g** reflects the main computation of the cell

Hochreiter & Schmidhuber (1997)

http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Some Example Slides

Computing Gradients: Backpropagation

$$\mathcal{L}(\mathbf{x}, i^*) = W\mathbf{z} \cdot e_{i^*} - \log \sum_{j=1}^{m} \exp(W\mathbf{z} \cdot e_j)$$

$\mathbf{z} = g(Vf(\mathbf{x}))$
Activations at hidden layer

- Gradient with respect to *V*: apply the chain rule

$$\frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial V_{ij}} = \frac{\partial \mathcal{L}(\mathbf{x}, i^*)}{\partial \mathbf{z}} \frac{\partial \mathbf{z}}{V_{ij}} \qquad \frac{\partial \mathbf{z}}{V_{ij}} = \frac{\partial g(\mathbf{a})}{\partial \mathbf{a}} \frac{\partial \mathbf{a}}{\partial V_{ij}} \qquad \mathbf{a} = Vf(\mathbf{x})$$

- First term: gradient of nonlinear activation function at ***a*** (depends on current value)
- Second term: gradient of linear function
- Straightforward computation once we have *err*(**z**)

Background Test

- Problem Set 0 (math background) is released, **due Thursday Aug 27**.
- Project 0 (programming - logistic regression) is also released, **due Sep 3**.
- Take **CS 4641/7641 Machine Learning** and (Math 2550 or Math 2551 or Math 2561 or Math 2401 or Math 24X1 or 2X51 - equivalent) before this class.
- If you want to understand the lectures better and complete homework with more ease, taking also CS 4644/7643 Deep Learning before this class.

Wait List

- If you plan to take the class, please complete and submit Problem Set 0 by Thursday Aug 27.
- If you get off the wait list, you will be automatically added to Gradescope after about a day. If not, post a private message on Ed Discussion to get the access to Gradescope.
- If you cannot access Gradescope by the due date, please email your submission to the instructor.
- Otherwise, unless specifically instructed to email, please do not use email to reaching the teaching team. Instead, post a private message on Ed Discussion. *(This applies to all students in the class.)*

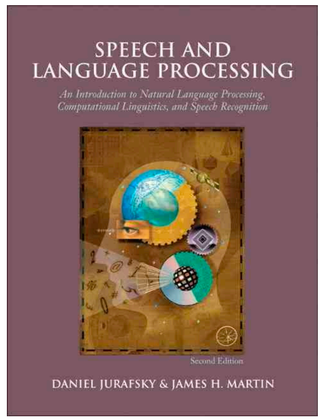
Free Textbooks!

- Two really awesome textbooks available
  - There will be assigned readings from both
  - Both freely available online

Speech and Language Processing (3rd ed. draft)

Dan Jurafsky and James H. Martin

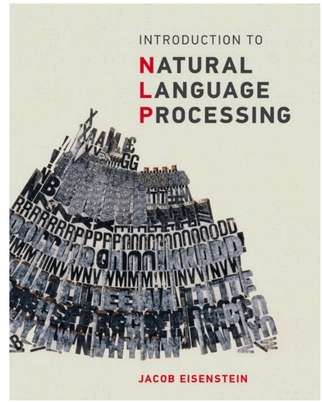
Here's our December 30, 2020 draft! Includes:



Introduction to Natural Language Processing

By Jacob Eisenstein

Published by The MIT Press
Oct 01, 2019 | 536 Pages | 7 x 9
| ISBN 9780262042840



Coursework Plan

- Four programming projects (28%)
  - Implementation-oriented
  - 1.5~2 weeks per assignment
  - fairly substantial implementation effort except P0
- Three written assignments (20%) + in-class midterm exam (20%)
  - Mostly math and theoretical problems related to ML / NLP
- Final project (25%)
- Attendance, In-class Presentation, and Participation (7%)

Programming Projects

- Four Programming Assignments (28% grade)

  - P0. Logistic regression (3%)
  - P1. Text Classification with Neural BOW, LSTM (5%)
  - P2. bidirectional LSTM-CNN (10%) + CRF (bonus)
  - P3. Fine-tuning + QLoRA (10%)

Programming Projects

- Modern NLP methods require non-trivial computation
  - Training/debugging neural networks can take a long time (**start early!**)
  - Most programming will be done with **PyTorch** library (can be tricky to debug)
  - You will want to use a GPU (Google Colab; free or pro account for $10/month)
  - The programming projects are designed with Google Colab in mind

Midterm

- Midterm date is set to **the week of Oct 26 and 28**.

- The midterm will be closed notes, books, laptops, smartphones, and people.

  You may (optionally) bring a calculator, if you want to.

- 75 minutes in class.

Midterm

- **Preparation:**
  - Lecture slides + textbook readings
  - Written homework (PS0, PS1, and PS2)
  - Programming Project 0, 1, 2, and 3

- **Expectation:**
  - **As a close-book close-note exam,** you are not expected to be able to answer all the questions correctly.
  - You may get challenged on some of the questions.
  - Try your best! We may curve up for final letter grades, if it is needed …

Final Project

- Final project (25%)
  - Groups of 2-4 student preferred, 1 student is also possible with permission.
    - 2-page proposal (team, literature review, proposed idea, preliminary results)
    - 4-5 page project report (similar to ACL/NAACL/EMNLP short papers: https://arxiv.org/search/?query=EMNLP+short+paper&searchtype=comments&source=header)
    - Final project presentation (video recording or in-person)
    - Good idea to run your project idea with me during office hour.

Final Project

- Grading rubrics
  - Clarity (1-5): For the reasonably well-prepared reader, is it clear what was done and why? Is the report well-written and well structured?
  - Originality / Innovativeness (1-5): How original is the approach? Does this project break new ground in topic, methodology, or content? How exciting and innovative is the work that it describes?
  - Soundness / Correctness (1-5): First, is the technical approach sound and well-chosen? Second, can one trust the claims of the report – are they supported by proper experiments, proofs, or other argumentation?
  - Meaningful Comparison (1-5): Does the author make clear where the problems and methods sit with respect to existing literature? Are any experimental results meaningfully compared with the best prior approaches?
  - Substance (1-5): Does this project have enough substance, or would it benefit from more ideas or results? Note that this question mainly concerns the amount of work; its quality is evaluated in other categories.
  - **Overall (1-5) - Overall quality/novelty/significance of the work. <u>Not</u> a sum of aspect-based scores.**

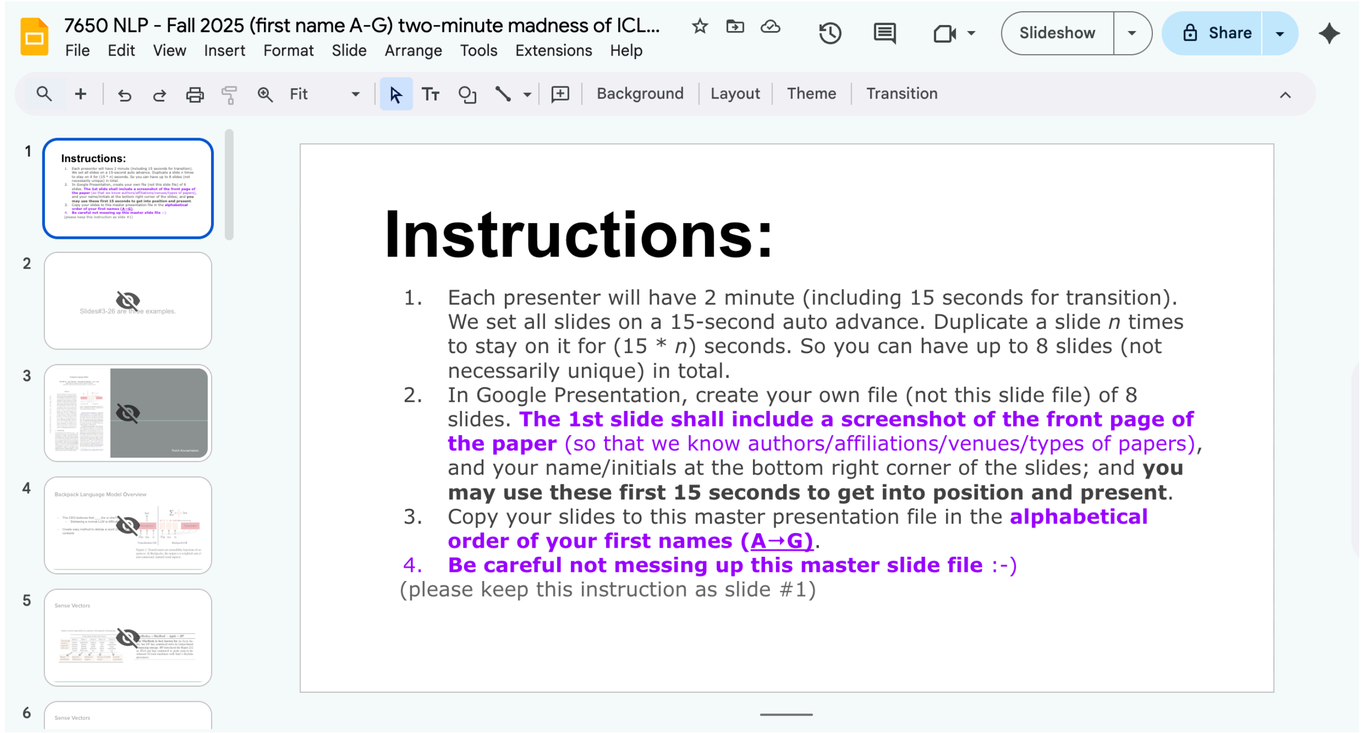
Final Project

- Rule of thumb — how much effort was put into the project?

  - Everyone comes with different backgrounds. We expect students will do different types of projects, some are more technical than others.
  - We most likely can tell how much effort you put into a project by looking at your project final report and presentation.
  - Creativity is also highly valued, and can take many forms:
    - What's new compared to existing work?
    - Will someone who reads your report learn something useful?

2-min Madness

- **In-class presentation of a recent research paper (2%)**
  - read recent publications (ACL/NAACL/EMNLP/ICLR/ICML/NeurIPS)
  - pick one and give a 2-minute oral presentation
    (detailed instructions on the submission will be released later)



Late Policy

- Late Policy
  - 6 flexible days to use over the duration of the semester for homework assignment only.
  - These flexible days should be reserved for emergency situation only.
  - Homework submitted late after all flexible days used up will receive penalty (5% deduction per day).

- No make-up exam for midterm. No late submission for final project report.
  - Unless under emergency situation verified by the Office of the Dean of Students

FAQ

- Q: The class is full, can I still get in?

  Depending on how many students will drop the class. The course registration system and office controls the process and priority order.

- Q: I am taking CS 4641/7641 ML class this same semester, would that be sufficient?

  A: No. You need to take 4641/7641 (or equivalent) **before** taking this class. NLP is at the very front of technology development. This is one of the most advanced classes. This course will be more work-intensive than most graduate or undergraduate courses at Georiga Tech, but will be comparable to NLP classes offered at other top universities.

- Q: How much grades I need to pass the class?

  A: Students need to receive 50% grade to pass the class.
  Students who choose Pass/Fail are not expected to participate in the group project.

FAQ

- Q: I want to understand the lectures better, what can I do?

  A: Read the required reading before the class. The two textbooks are both awesome! Taking deep learning class first will greatly help too. The lectures are designed to cover state-of-the-art material in class, while lower-level details will be "taught" through written and programming homework assignments.
  (similar design to NLP classes at other top universities, e.g., Stanford/Berkeley/Princeton)

- Q: I want to learn more about LLMs, what can I do?

  A: Read some recent papers (last component of this class), and take CS 7652 "Large Language Models"

  See also — CS 8803-LLM "Large Language Model"
  https://cocoxu.github.io/CS8803-LLM-fall2024/ (Fall 2024)
  https://cocoxu.github.io/CS8803-LLM-spring2026/ (Spring 2026)

QA Time