Advanced NLP & AI for Linguistics
Attention Probing, Causality, and Interpretability

LING 7XXX – Spring 2027

Course information

Time	TBD
Location	TBD
Instructor	Utku Turk
Email	utkuturk@umd.edu
Office hours	By appointment
Course site	Canvas

Prerequisites: Background in neural networks (e.g., intro ML for language) or permission. Familiarity with Python and PyTorch helpful.

1 Course description

What do neural language models actually learn? How can we probe their internal representations? Can we establish causal relationships between model components and linguistic behavior?

Neural networks are often treated as black boxes. In this seminar, we will open them up together. This advanced seminar explores cutting-edge methods for interpreting and analyzing neural networks through a linguistic lens. We focus on three core areas: (i) **attention probing**—analyzing attention patterns to understand how models process syntactic dependencies, (ii) **causality and intervention**—using counterfactual methods to establish causal relationships, and (iii) **representation analysis**—extracting and interpreting hidden representations for linguistic knowledge.

Students will learn to design rigorous probing experiments, implement causal intervention methods, analyze attention entropy and head specialization, and critically evaluate interpretability claims. This is a research-oriented seminar where students develop original proposals for investigating what neural models learn about language.

Background in syntax/semantics strongly recommended.

2 Learning objectives

Upon successful completion of this course, students will be able to:

1. Design and implement probing classifiers to test for linguistic knowledge
2. Analyze attention patterns for syntactic dependencies and semantic roles
3. Apply causal intervention methods (counterfactual data augmentation, interchange interventions)
4. Compute and interpret attention entropy and head specialization metrics
5. Extract and visualize hidden representations using dimensionality reduction
6. Critically evaluate interpretability claims and methodological choices
7. Propose original research investigating model representations
8. Read and present cutting-edge papers on neural network interpretability

3 Course structure

Weekly rhythm

Component	What it looks like
Paper discussion	Student-led presentation + group discussion of 1–2 papers.
Methods tutorial	Hands-on coding session implementing the week's method.
Research clinic	Workshopping student project ideas, designs, and analyses.

Tools and methods

- **Probing:** Linear probes, control tasks, diagnostic classifiers
- **Attention analysis:** Attention weights, entropy, rollout, head pruning
- **Causality:** Counterfactual data augmentation, causal mediation analysis, interchange interventions
- **Representation analysis:** PCA, t-SNE, UMAP, CKA similarity
- **Frameworks:** Hugging Face Transformers, AllenNLP Interpret, Captum

4 Course requirements

4.1 Grading

Item	%	What counts
Participation	15	Active discussion; constructive feedback on peer presentations.
Paper presentations (2x)	20	Lead discussion on 2 papers (10% each): summarize, critique, propose extensions.
Method implementations (3x)	30	Implement and document 3 methods from class (10% each).
Final project	35	Full implementation: research question, experiments, analysis, paper, code, presentation.

4.2 Paper presentations

Each student will lead discussion on 2 papers during the semester. Your presentation should:

- Summarize the research question, method, and findings (15 min)
- Critique the methodology: what assumptions are made? What controls are missing?
- Propose 2–3 extensions or follow-up experiments
- Prepare 3–5 discussion questions for the class

4.3 Method implementations

Choose 3 methods from the course to implement and document. Each implementation should include:

- Working code (well-commented Colab notebook or script)
- Brief write-up (2–3 pages): what the method does, when to use it, example application
- Demonstration on a small dataset or pretrained model

4.4 Final project

Conduct an original research project using probing, causality, or interpretability methods to investigate a linguistic question. This is a full implementation project that should include:

- Research question and linguistic motivation
- Literature review of relevant interpretability work

- Implementation of your method (probing task, intervention design, or analysis technique)
- Experiments on real models and datasets
- Systematic evaluation and analysis of results
- Discussion of what you found and theoretical implications
- Limitations and future directions

Format: 10–12 page paper in ACL style + code repository + 15-minute presentation.

Important: Your project must be grounded in serious theoretical linguistics, not just ML applications. The research question should emerge from existing work in syntax, semantics, phonology, psycholinguistics, or sociolinguistics. Examples of appropriate projects:

- Testing feature bundle representations in Distributed Morphology or nanosyntax frameworks
- Quantifying language contact patterns (e.g., Balkan Sprachbund vs. Anatolian minority languages)
- Analyzing power relations in indigenous languages of the Americas using computational methods
- Probing for ellipsis licensing conditions and information structure
- Testing syntactic theories (Minimalism, HPSG, LFG) through model representations
- Cross-linguistic variation in agreement systems (following Polinsky, Preminger, etc.)
- Encoding semantic primitives (Kratzer, Heim, Chierchia) in neural representations

See recent SCiL proceedings for examples of theory-driven computational work.

5 Course schedule (16-week semester)

Schedule subject to change. Canvas is the live version.

Wk	Topic	Readings	Methods / due
1	Introduction; levels of analysis	**Req:** Newell (1973); Marr (1982) [Ch. 1]; Belinkov & Glass (2019) "Analysis methods in neural NLP". **Opt:** Linzen & Baroni (2021) "Syntactic structure from deep learning".	Setup: Hugging Face, extracting representations
2	Probing classifiers I: methodology	**Req:** Hewitt & Liang (2019) "Designing and interpreting probes"; Conneau et al. (2018) "What you can cram into a single vector". **Opt:** Belinkov (2022) "Probing classifiers"; Warstadt et al. (2020) "Linguistic analysis of pretrained sentence encoders with acceptability judgments".	Implementing linear probes
3	Probing classifiers II: control tasks	**Req:** Hewitt & Manning (2019) "A structural probe for finding syntax"; Pimentel et al. (2020) "Information-theoretic probing". **Opt:** Voita & Titov (2020) "Information-theoretic probing with MDL"; White et al. (2021) "A non-parametric test to detect syntactic representations in neural language models".	Control tasks; selectivity metrics
4	Attention analysis I: patterns & dependencies	**Req:** Clark et al. (2019) "What does BERT look at?"; Vig & Belinkov (2019) "Analyzing attention in transformers". **Opt:** Htut et al. (2019) "Do attention heads track syntactic dependencies?"; Kulmizev et al. (2020) "Do neural language models show preferences for syntactic filler–gap dependencies?"	Visualizing attention; extracting patterns **Due:** Method 1
5	Attention analysis II: entropy & specialization	**Req:** Voita et al. (2019) "Analyzing multi-head self-attention"; Michel et al. (2019) "Are sixteen heads really better than one?" **Opt:** Kovaleva et al. (2019) "Revealing the dark secrets of BERT".	Computing attention entropy; head pruning
6	Causal intervention I: counterfactuals	**Req:** Kaushik et al. (2020) "Learning the difference that makes a difference"; Gardner et al. (2020) "Evaluating models' local decision boundaries". **Opt:** Wu et al. (2021) "Polyjuice: Generating counterfactuals"; Finlayson et al. (2021) "Causal analysis of syntactic agreement mechanisms in neural language models".	Counterfactual data augmentation
7	Causal intervention II: mediation analysis	**Req:** Vig et al. (2020) "Investigating gender bias in language models using causal mediation analysis"; Finlayson et al. (2021) "Causal analysis of syntactic agreement mechanisms". **Opt:** Geiger et al. (2021) "Causal abstractions of neural networks".	Causal mediation analysis; interchange interventions **Due:** Method 2
8	Representation geometry I: similarity & clustering	**Req:** Saphra & Lopez (2019) "Understanding learning dynamics of language models"; Kornblith et al. (2019) "Similarity of neural network representations revisited". **Opt:** Ethayarajh (2019) "How contextual are contextualized word representations?"	CKA similarity; representation clustering
9	Representation geometry II: subspaces	**Req:** Ravfogel et al. (2020) "Null it out: Guarding protected attributes by iterative nullspace projection"; Bolukbasi et al. (2016) "Man is to computer programmer as woman is to homemaker?" **Opt:** Hewitt & Manning (2019) "A structural probe for finding syntax" [revisit].	Subspace methods; INLP
10	Cross-linguistic probing	**Req:** Pires et al. (2019) "How multilingual is multilingual BERT?"; Chi et al. (2020) "Finding universal grammatical relations in multilingual BERT". **Opt:** Wu & Dredze (2020) "Are all languages created equal in multilingual BERT?"; Ravfogel et al. (2019) "Studying the inductive biases of RNNs with synthetic variations of natural languages".	Probing multilingual models **Due:** Method 3
11	Encoding linguistic structure	**Req:** Tenney et al. (2019) "BERT rediscovers the classical NLP pipeline"; Jawahar et al. (2019) "What does BERT learn about the structure of language?" **Opt:** Liu et al. (2019) "Linguistic knowledge and transferability of contextual representations"; Wilcox et al. (2018) "What do RNN language models learn about filler–gap dependencies?"	Edge probing; layer-wise analysis
12	Limitations of interpretability methods	**Req:** Belinkov & Glass (2019) "Analysis methods in neural NLP" [full]; Jacovi & Goldberg (2020) "Towards faithfulness and consistency in explanations". **Opt:** Lyu et al. (2021) "Towards faithful model explanation in NLP".	Critique workshop; alternative explanations
13	Emergent linguistic abilities	**Req:** Wei et al. (2022) "Emergent abilities of large language models"; Schaeffer et al. (2023) "Are emergent abilities of LLMs a mirage?" **Opt:** Lampinen et al. (2022) "Can language models learn from explanations in context?"	Testing for emergence; scaling analysis
14	Mechanistic interpretability	**Req:** Olsson et al. (2022) "In-context learning and induction heads"; Elhage et al. (2021) "A mathematical framework for transformer circuits". **Opt:** Wang et al. (2023) "Interpretability in the wild".	Circuit analysis; activation patching
15	Student project presentations I	–	10-minute presentations; peer feedback
16	Student project presentations II	–	10-minute presentations; peer feedback
Finals	–	–	**Due:** Final project proposal

6 Policies

6.1 What you might struggle with (and how to succeed)

This is an advanced seminar with high expectations. Here's what students typically struggle with and evidence-based advice:

Time management:

- Expect 10–12 hours/week outside class: 4–5 hours reading papers deeply, 3–4 hours implementing methods, 2–3 hours on project
- Reading papers takes time: Budget 2–3 hours per paper for deep reading, note-taking, and critique
- Start implementations early: Debugging interpretability methods is harder than standard ML

Reading research papers:

- Read strategically: Abstract → conclusions → figures → introduction → methods → results
- Take notes by hand: Handwriting improves retention and critical thinking (Mueller & Oppenheimer, 2014)
- For each paper, write: (1) Main claim, (2) Evidence, (3) Limitations, (4) Your critique, (5) Follow-up questions
- Don't use note-taking apps during reading: They encourage shallow processing and procrastination

Presenting papers:

- Understand deeply, don't just summarize: Run the code if available, try to break the method
- Prepare critiques: What assumptions are made? What controls are missing? What would you do differently?
- Practice your presentation: 15 minutes goes fast

Implementing methods:

- Start with toy examples: Test on small data before scaling up
- Document as you go: Future you will thank present you
- Compare to paper's results: If your implementation doesn't match, debug systematically

Note-taking in seminar:

- Take notes by hand, not laptop: Forces active processing (Mueller & Oppenheimer, 2014)
- Why handwriting works: Can't transcribe verbatim, must synthesize and connect ideas
- After class: Transfer key insights to digital notes for organization

Project development:

- Start early: Finding the right research question takes time
- Iterate: First idea rarely works; plan for 2–3 pivots
- Get feedback: Use research clinics to workshop ideas

Resources:

- How to Read a Paper (Keshav, 2007): `https://web.stanford.edu/class/ee384m/Handouts/HowtoReadPaper.pdf`
- How to Read a Research Paper (Mitzenmacher, 2016): `https://www.eecs.harvard.edu/~michaelm/postscripts/ReadPaper.pdf`
- The Science of Effective Learning (Dunlosky et al., 2013): `https://doi.org/10.1177/1529100612453266`
- Writing Reviews for NLP Conferences: `https://aclrollingreview.org/reviewertutorial`

6.2 Attendance

Attendance is expected. This is a discussion-based seminar; your participation is essential.

6.2 Academic integrity

Do your own work; cite sources. Collaboration on method implementations is encouraged, but write-ups must be your own.

6.3 Use of LLMs

LLMs may be used for coding assistance and brainstorming, but not for writing paper summaries or project proposals. Document any use.

6.4 Accessibility & wellness

Contact the relevant office for accommodations. Reach out if you're struggling—we can make a plan.

7 Resources

Key papers & tutorials

- Belinkov & Glass (2019). "Analysis methods in neural language processing: A survey." *TACL*.
- Rogers et al. (2020). "A primer on neural network models for natural language processing." *JAIR*.
- Doshi-Velez & Kim (2017). "Towards a rigorous science of interpretable machine learning."

Technical resources

Category	Links
Frameworks	Hugging Face Transformers `https://huggingface.co/`; AllenNLP Interpret `https://allennlp.org/interpret`
Causality	Captum `https://captum.ai/`; InterpretML `https://interpret.ml/`
Visualization	BertViz `https://github.com/jessevig/bertviz`; exBERT `https://exbert.net/`
Probing	SentEval `https://github.com/facebookresearch/SentEval`

This syllabus is a living document and may be updated during the semester. Last updated: January 3, 2026

Appendix: Quarter-system (10 weeks)

Wk	Topic	Readings	Methods / due
1	Intro; probing classifiers	**Req:** Newell (1973); Marr (1982) [Ch. 1]; Hewitt & Liang (2019); Conneau et al. (2018).	Linear probes; control tasks
2	Attention analysis	**Req:** Clark et al. (2019); Voita et al. (2019); Michel et al. (2019).	Attention patterns; entropy; head pruning **Due:** Method 1
3	Causal interventions	**Req:** Kaushik et al. (2020); Vig et al. (2020); Finlayson et al. (2021).	Counterfactuals; mediation analysis
4	Representation geometry	**Req:** Saphra & Lopez (2019); Kornblith et al. (2019); Ravfogel et al. (2020).	CKA similarity; subspace methods **Due:** Method 2
5	Cross-linguistic & multilingual	**Req:** Pires et al. (2019); Chi et al. (2020); Tenney et al. (2019).	Probing multilingual models
6	Encoding linguistic structure	**Req:** Jawahar et al. (2019); Liu et al. (2019); Belinkov & Glass (2019) [full].	Edge probing; layer-wise analysis **Due:** Method 3
7	Limitations & mechanistic interpretability	**Req:** Jacovi & Goldberg (2020); Olsson et al. (2022); Elhage et al. (2021).	Circuit analysis; activation patching
8	Emergent abilities & scaling	**Req:** Wei et al. (2022); Schaeffer et al. (2023).	Testing for emergence
9	Project presentations I	–	10-min presentations
10	Project presentations II	–	10-min presentations
Finals	–	–	**Due:** Final proposal

Quarter grading:

Item	%	What counts
Participation	15	Active discussion.
Paper presentations (2x)	20	Lead discussion (10% each).
Method implementations (3x)	30	Code + write-up (10% each).
Final project	35	Full implementation: paper + code + presentation.