AIM 5014 Special Topic: Advanced NLP (3 credits)

Adam Faulkner

Fall 2025

✉ adam.faulkner@yu.edu
Course Github site

Class hours: Tuesdays 5:30-7:30pm
Class Room: SCW - 245 Lexington Ave. Rm 101

Course Description

Large Language Models (LLMs) such as OpenAI's ChatGPT, Google's Gemini, Deepseek, and Meta's LLaMA have transformed the field of NLP and the generative capabilities of these models are being rapidly integrated into everyday-use applications such as search engines, text editors, and integrated development environments. This course offers students a comprehensive introduction to the world of LLMs. In addition to foundational topics related to LLMs such as the Attention mechanism, self-training, and finetuning, students will gain a thorough grounding in emerging topics in LLMs such as Retrieval Augmented Generation (RAG), Agents, defenses against adversarial attacks, LLM-as-a-Judge, LLM interpretability, and hallucination detection.

Course Objectives

By the end of this course students will have

- gained a theoretical understanding of the Transformer and its secret sauce, the Attention mechanism, as well the Transformer's three major variants: the encoder-decoder, encoder-only, and decoder-only architectures
- gained hands-on-experience finetuning and prompting open-source small LMs, such as BERT, and LLMs such as Meta's LLaMA3 as well as experience integrating open-source LLMs into popular LLM paradigms such as RAG and the Agent framework
- gained an understanding of postraining frameworks such as Direct-Preference Optimization and Deepseek's Group Relative Policy Optimization
- learned advanced prompting techniques such as ReAct and CodeAct and tested these techniques in NLP tasks such as dialogue and summarization using popular LLM libraries such as *LangChain*

- gained an understanding of RAG, tool-retrieval, and the Agent paradigm as well as experience implementing each of these in downstream applications using LangGraph
- learned about common adversarial, or "jailbreaking" attacks against LLMs and defenses against these attacks
- learned about contemporary research into LLMs' ability to utilize uniquely human cognitive traits such as theory of mind, logical induction, and linguistic competence

Course Material

The material for this course consists of

- representative research papers or textbook chapters containing technical presentations of each week's topic. All of this material is linked in the Course Schedule section below.
- Jupyter notebooks illustrating implementations of the concepts described in each week's topic. These notebooks can be reviewed before each class and run on Collab.
- slides presented in class. These will also be made available on the course Github site.

Assignments

Students will complete 5 assignments. These are short, non-programming assignments that delve into the math underlying the concepts presented in the lectures. These assignments must be printed, completed by hand, and submitted online (via scan or photo) or handed in to me directly.

The final group project can be completed singly or as a group (max 5 members). The project should consist of an LLM-based application that implements one or more of the concepts described in class. The application will be demoed on the last day of class as a 5 -10 minute presentation.

Grading

- **10%** of the student's grade will be determined by attendance and participation in class
- **70%** of the student's grade will be determined by 6 take-home math assignments.
- **20%** of the student's grade will be determined by a final group project.

Course Schedule

Class 1 Aug 26

Course introduction

Class 2 Sept 2

Neural Networks & Deep Learning

- Perceptrons
- Deep Feedforward Neural Networks
- Gated Architectures: RNNs, LSTMs

Libraries: *PyTorch*

- Goodfellow, Bengio, & Courville: Introduction
- Jurafsky & Martin Chapter 7: Neural Networks and Neural Language Models
- Jurafsky & Martin Chapter 9: RNNs and LSTMs

Class 3 Sept 9

Early Language Modeling

- N-gram-models
- Neural architectures
- Embeddings
- Applications of embeddings

Libraries: *PyTorch*, *Gensim*

- Jurafsky & Martin Chapter 3: Ngram Language Models
- Jurafsky & Martin Chapter 6: Vector Semantics & Embeddings
- Efficient Estimation of Word Representations in Vector Space

Class 4 Sept 16

Assignment 1 Due

The Transformer

- Attention
- Self-attention
- Multi-Head Attention
- Sparse Attention

Encoder-Decoder architectures

Libraries: *Hugging Face*

- Neural Machine Translation by Jointly Learning to Align and Translate
- Attention is all you need
- Generating Long Sequences with Sparse Transformers
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Sept 23: No class

Class 5 Sept 30

Variant Transformer architectures and an introduction to finetuning

- Encoder-only architectures
- Decoder-only architectures
- The finetuning paradigm

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 11: Masked Language Models
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Language Models are Unsupervised Multitask Learners
- Language Models are Few-Shot Learners

Oct 7: No class

Assignment 2 Due

Oct 14: No class

Class 6 Oct 21

LLMs: Data, modeling, and tokenization

- Finetuning MLMs for NLP tasks
- Casting NLP tasks as word-prediction tasks
- Data
- Tokenization
    - WordPiece
    - Byte-Pair encoding

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 10: Large Language Models
- The Pile: An 800GB Dataset of Diverse Text for Language Modeling
- Masked language modeling (HF tutorial)
- Causal language modeling (HF tutorial)
- Fast WordPiece Tokenization
- Neural machine translation of rare words with subword units.

Class 7 Oct 28

Assignment 3 Due

LLMs: Aligning LLMs to human preferences and instructions

- Reinforcement Learning from Human Feedback
- Direct Preference Optimization
- Mixture-of-Experts

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 12: Model Alignment, Prompting, and In-Context Learning
- Training language models to follow instructions with human feedback
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model
- Mixtral of Experts

Class 8 Nov 4

LLM pretraining and inference

- Quantization
- Parameter-efficient finetuning

Libraries: *Hugging Face*

- QLORA: Efficient Finetuning of Quantized LLMs
- Quantization, QLora (HF post)

Class 9 Nov 11

Augmented LLMs

- Self-reflection
- Retrieval-Augmented Generation
- The Agent Framework
- Multi-Agent frameworks
- The CodeAct framework

Libraries: *FAISS*, *LangChain*, LangGraph

- ReAct: Synergizing Reasoning and Acting in Language Models
- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
- RAGAS: Automated Evaluation of Retrieval Augmented Generation
- Agents (HF tutorial)
- Exploring Large Language Model Based Intelligent Agents: Definitions, Methods, and Prospects
- Executable Code Actions Elicit Better LLM Agents

Class 10 Nov 18

Assignment 4 Due

Automated prompt engineering and LLM Evaluation

- Prompt tuning
- LLM Evaluation
- Model Steering

- AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts
- RLPROMPT: Optimizing Discrete Text Prompts with Reinforcement Learning
- Evaluating Large Language Models: A Comprehensive Survey

Class 11 Nov 25

Cutting-edge architectures: DeepSeek and GPT 5

- Long context extension with YaRN
- Multi-head Latent Attention
- DeepSeekMoE (Deep Seek Mixture of Experts)
- Multi-token prediction training
- Group-Relative Policy Optimization

- DeepSeek tutorial
- DeepSeek v1 paper
- DeepSeek v3 paper
- GPT-OSS Open AI post

Class 12 Dec 2

LLM interpretability and hallucination detection

- LLM Interpretability
- LLM hallucination detection and remediation

- Rethinking Interpretability in the Era of Large Language Models
- Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models
- Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection
- SELFCHECKGPT: Zero-Resource Black-Box Hallucination Detection for Generative Large Language Models
- Language Models Implement Simple Word2Vec-style Vector Arithmetic

Class 13 Dec 9

Assignment 5 Due

Jailbreaking LLMs: Attacks and defenses

- Adversarial Attacks
- Guardrails
- Unlearning

Libraries: *NeMo-Guardrails (NVIDIA)*

- Extracting Training Data from Large Language Models
- Threats to Pre-trained Language Models: Survey and Taxonomy
- Llama Guard: LLM-based Input-Output Safeguard for Human-AI Conversations
- NeMo Guardrails: A Toolkit for Controllable and Safe LLM Applications with Programmable Rails
- Large Language Model Unlearning

Class 14 Dec 16

Additional topics in LLMs

- Detection of LLM-generated text
- Machine Psychology
- What do LLMs Really Understand?

- Can AI-generated Text be Reliably Detected?
- Ghostbuster: Detecting Text Ghostwritten by Large Language Models
- On the dangers of stochastic parrots: Can language models be too big?
- Machine Psychology
- The debate over understanding in AI's large language models
- Do Prompt-Based Models Really Understand the Meaning of Their Prompts?
- Dissociating Language and Thought in Large Language Models
- Modern language models refute Chomsky's approach to language

Class 15 Dec 23

Final project presentations