AIM 5011-1 Natural Language Processing (3 credits)

Adam Faulkner

Spring 2026

✉ adam.faulkner@yu.edu
Course Github site

Class hours: Tuesdays 5:30-7:30pm
Classroom: 215 Lexington Avenue LX312

Course Description

This course provides a comprehensive overview of Natural Language Processing, the sub-field of Artificial Intelligence that deals with the automated processing of natural language text. The course is divided into three sections. In the first section, we'll introduce classic approaches to modeling natural language using traditional machine learning techniques such as Logistic Regression, Naive Bayes, and n-gram language modeling, as well as sparse vector-based representations of text such as bag-of-words, before moving to more contemporary, Deep Learning-based modeling techniques and vector representations such as LSTMs and dense-vector representations. Section 1 ends with the introduction of the Transformer and describes the field's move to language-model-based solutions to NLP tasks via frameworks such as finetuning.

Section 2 provides an overview of traditional NLP tasks such as classification (Sentiment Analysis, Intent Detection, etc.), sequence labeling (Named Entity Recognition, syntactic and semantic parsing, etc.), and text generation (Machine Translation, Question-Answering, Summarization, etc.).

Section 3 introduces Large Language Models (LLMs), the dominant paradigm of contemporary NLP. We'll learn how LLMs reframed the NLP tasks described in section 2, which were traditionally solved using the ML techniques described in section 1, as word-prediction tasks. In addition to foundational topics related to LLM such as the autoregressive language modeling objective underlying GPT-style models and post-training regimes such as Reinforcement Learning from Human Feedback, students will gain a thorough grounding in emerging topics in LLMs such as Retrieval Augmented Generation (RAG), Agents, LLM-as-a-Judge, LLM interpretability, and hallucination detection.

Course Objectives

By the end of this course students will have

- gained an understanding of traditional ML approaches to solving NLP tasks as well as early language modeling techniques such as n-gram models
- gained an understanding of the tradeoffs between traditional, sparse-vector-based representations of text and more contemporary, dense-vector-based representations
- gained a theoretical understanding of the Transformer as well the Transformer's three major variants: the encoder-decoder, encoder-only, and decoder-only architectures
- gained an understanding of the three major flavors of NLP tasks: classification, sequence labeling, and text generation
- 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 RAG, tool-retrieval, and the Agent paradigm
- gained an understanding of LLM interpretability and LLM security risks and remediation

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 a combination of math and programming. Math assignments must be printed, completed by hand, and submitted online (via scan or photo) or handed in to me directly. The programming assignments can be completed as Jupyter notebooks and uploaded to Canvas.

The final group project can be completed singly or as a group (max 5 members). The project should consist of an NLP 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 5 take-home math assignments.
- **20%** of the student's grade will be determined by a final group project.

Course Schedule

Class 1 Jan 20

Course Introduction & Early text classification

- Logistic Regression
- The bag-of-words vector representation
- The Term-Document Matrix representation

Libraries: *sklearn*

- Jurafsky & Martin Chapter 4:Logistic Regression & Text Classification

Class 2 Jan 27

Early Language Modeling

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

Libraries: *sklearn, gensim*

- Jurafsky & Martin Chapter 3: Ngram Language Models
- Jurafsky & Martin Chapter 5: Embeddings
- Efficient Estimation of Word Representations in Vector Space

Class 3 Feb 3

Assignment 1 Due

Neural Networks

- Perceptrons
- Deep Feedforward Neural Networks

Libraries: *PyTorch*

- Goodfellow, Bengio, & Courville: Introduction
- Jurafsky & Martin Chapter 6: Neural Networks

Class 4 Feb 10

Gated Architectures: RNNs and LSTMs

- RNNs
- LSTMs

Libraries: *PyTorch*

- Jurafsky & Martin Chapter 13: RNNs and LSTMs

Class 5 Feb 17

The Transformer

- Attention
- Self-attention
- Multi-Head Attention
- Sparse Attention
- Encoder-Decoder architectures

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 8: Transformers
- Attention is all you need
- Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Class 6 Feb 24

Assignment 2 Due

Variant Transformer architectures and an introduction to finetuning

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

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 10: Masked Language Models
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding
- Language Models are Few-Shot Learners

March 3: No class

Class 7 March 10

Assignment 3 Due

Classification tasks

- Sentiment Analysis
- Argumentation Mining
- Entailment
- Intent Detection and Slot Filling
- Coreference resolution

Libraries: *Hugging Face, NLTK, spaCy*

- Jurafsky & Martin Chapter 4:Logistic Regression & Text Classification
- Deep Learning in Stance Detection: A Survey
- Argument Mining: A Survey
- Jurafsky & Martin Chapter 23: Coreference Resolution and Entity Linking

Class 8 March 17

Sequence labeling tasks

- Named Entity Recognition
- Syntactic parsing
- Semantic parsing

Libraries: *Hugging Face, NLTK, spaCy*

- Jurafsky & Martin Chapter 17: Sequence Labeling for Parts of Speech and Named Entities
- Jurafsky & Martin Chapter 18:Context-Free Grammars and Constituency Parsing
- Jurafsky & Martin Chapter 21: Semantic Role Labeling

Class 9 March 24

Assignment 4 Due

Text generation tasks

- Machine Translation
- Question-Answering
- Summarization
- Paraphrase generation
- Commonsense reasoning

Libraries: *Hugging Face*

- Jurafsky & Martin Chapter 12: Machine Translation
- A Systematic Survey of Text Summarization: From Statistical Methods to Large Language Models
- Paraphrase Generation: A Survey of the State of the Art
- Commonsense Knowledge Reasoning and Generation with Pre-trained Language Models: A Survey

Class 10 March 31

Large Language Models: 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 7: Large Language Models
- The Pile: An 800GB Dataset of Diverse Text for Language Modeling
- Masked language modeling (HF tutorial)
- Causal language modeling (HF tutorial)

April 7: No class

Class 11 April 14

Assignment 5 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 9: Post-training: Instruction Tuning, Alignment, and Test-Time Compute
- Training language models to follow instructions with human feedback
- Direct Preference Optimization: Your Language Model is Secretly a Reward Model
- Mixtral of Experts

Class 12 April 21

Augmented LLMs

- Self-reflection
- Retrieval-Augmented Generation
- The Agent Framework

Libraries: *FAISS, LangChain, LangGraph*

- ReAct: Synergizing Reasoning and Acting in Language Models
- Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks
- Agents (HF tutorial)
- Exploring Large Language Model Based Intelligent Agents: Definitions, Methods, and Prospects

Class 13 April 28

LLM interpretability and hallucination detection

- LLM Interpretability
- LLM hallucination detection and remediation

Libraries: *Hugging Face*

- Rethinking Interpretability in the Era of Large Language Models
- Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models

Class 14 May 5

Additional topics in LLMs

- Detection of LLM-generated text
- Guardrails
- LLM fingerprinting
- Metacognition and deception

- Can AI-generated Text be Reliably Detected?
- DetectGPT: Zero-Shot Machine-Generated Text Detection using Probability Curvature
- Guardrails for Large Language Models: A Review of Techniques and Challenges

Class 15 May 12

Final project presentations