IPHS 391: Frontiers in Generative AI

Jon Chun
Kenyon College
Fall 2025
Wednesday Evenings 7pm-10pm
Location: Timberlake #5 (Evans Conference Rm)
Office Hours: TuTh 10:30-noon (2 Timberlake House)
Credit: 0.5/4, Section: 00, CRN: 80637

Fees: No textbooks, but up approximately $180 in cloud subscriptions

- $20/mo OpenAI ($80/4 mo)
- $10/mo Google Colab $40/4 mo)
- $20/mo Anthropic.ai ($40/2 mo)
- (1yr free) Google Gemini Pro Plan
    - https://gemini.google/students/
- (1mo free) Perplexity Pro
    - https://www.perplexity.ai/help-center/en/articles/10964633-student-referrals

SYLLABUS

DESCRIPTION:

This upper-division course offers an in-depth exploration of advanced AI concepts, focusing on interdisciplinary applications of large language models, AI information systems, and autonomous agents. Over 15 weeks, students will engage with a progressive curriculum, starting with a review of Python and a series of four hands-on projects: (a) OpenAI API programming a GPT-based chatbot, (b) mechanistic interpretations of transformer internals using Huggingface Transformers, (c) Retrieval-Augmented Generation (RAG) using LangChain, and (d) simulations of autonomous multi-agent systems using AutoGen. The course includes four substantive subprojects and one final project, enabling students to apply theoretical knowledge to practical, real-world AI challenges. This course is designed to equip students with the skills and knowledge necessary to innovate in the rapidly evolving field of artificial intelligence, emphasizing both technical proficiency and ethical considerations. Introductory Python programming experience required.

NOTE: These **4 broad frontiers of AI research** are **rapidly evolving** and based upon my AI research and industry consulting with Meta, IBM, the Whitehouse/NIST AI Safety Institute, etc. There is a constant flow of major new AI research, libraries, frameworks and startups nearly every week. Since this course will begin **9 months** after this syllabus was written, **expect**

updates to reflect the most recent in **AI research breakthroughs, tooling, and industry best practices as of August 2024**. Nonetheless, the class will be structured around these 4 broad and relatively consistent universal areas in AI.

SCHEDULE:

Week 1-3: OpenAI API Function Calling App

- **Week 1:** Course Introduction and AI Overview
- **Week 2:** Python and OpenAI API Basics
- **Week 3: Subproject 1:** Develop a basic function-calling application using OpenAI API
- *Semester Project Kickoff: Identifying a problem for AI application*

Week 4-6: Embeddings and Explainability

- **Week 4:** Introduction to GPT, Embeddings, and Language Models
- **Week 5:** Application of Embeddings in AI
- **Week 6: Subproject 2:** Exploring Embeddings and Explainability on a familiar dataset/task
- *Semester Project Check-in: Conceptual Design and Initial Research*

Week 7-9: RAG Application using LangChain

- **Week 7:** Introduction to Retrieval-Augmented Generation (RAG)
- **Week 8:** Working with LangChain for RAG Applications
- **Week 9: Subproject 3:** Implementing a RAG application using LangChain
- *Semester Project Check-in: Developing AI Models and Prototyping*

Week 10-12: Autonomous Multi-Agent Simulations and Benchmarking

- **Week 10:** Introduction to Autonomous Multi-Agent Systems
- **Week 11:** Simulation and Modelling of Multi-Agent Systems
- **Week 12: Subproject 4:** Creating and Benchmarking Autonomous Multi-Agent Simulations
- *Semester Project Check-in: Integration and Testing of AI Solutions*

Week 13-15: Finalization and Presentation of Semester Project

- **Week 13-14:** Final Development, Testing, and Refinement of Semester Projects
- **Week 15:** Final Project Presentations and Course Wrap-up

Each subproject is designed to reinforce the theoretical concepts taught in the respective weeks and to provide practical experience. The semester project spans the entire course, allowing students to gradually develop and refine a comprehensive AI application, culminating in a final presentation that showcases their learning and innovation.

GRADING:

Class Participation: 20%
Weekly Quizzes: 30%
4 Mini-Projects: 30%
1 Final Main Project: 20%

ASSIGNMENT DESCRIPTIONS:

Description of the 4 Mini Projects

Mini Project 1: OpenAI API Function Calling Chat App

- **Libraries/Frameworks Used:** Python, OpenAI's GPT-3 API
- **Project Description:** Students will develop an application that interfaces with OpenAI's API to create a simple chatbot with distinct persona, short-term and long-term memories and structured JSON function calls.
- **Interdisciplinary Examples:** Creating a chatbot for mental health support, developing a content generator for social media fake news, automating data summaries for research papers.
- **Grading Rubric:**
    - Functionality (30%): Application meets all specified requirements.
    - Creativity and Innovation (20%): Originality in application concept and execution.
    - Code Quality (20%): Clean, well-documented, and efficient code.
    - API Integration (20%): Effective use of OpenAI's API functionalities.
    - Presentation (10%): Clarity and completeness in explaining the project.

Mini Project 2: Embeddings and Explainability with Huggingface

- **Libraries/Frameworks Used:** Python, Specialized Embedding Models, and Huggingface API
- **Project Description:** Students will utilize embeddings in a machine learning model and implement explainability methods to interpret the model's decisions.
- **Interdisciplinary Examples:** Analyzing social media trends for sociological insights, interpreting musical embeddings for thematic structure, comparing English with under resourced language for semantic space anomalies.
- **Grading Rubric**
- Functionality (30%): Successful implementation of embeddings in the model.
    - Explainability and Interpretation (25%): Effective use of methods to interpret and explain model decisions.
    - Application Relevance (20%): Relevance and impact of the project in an interdisciplinary context.
    - Code Quality (15%): Cleanliness, organization, and documentation of code.

    - Presentation and Report (10%): Clear explanation of methodology, results, and interdisciplinary relevance.

Mini Project 3: RAG Application using LangChain

- **Libraries/Frameworks Used:** Python, LangChain, OpenAI's GPT-3 API
- **Project Description:** Students will develop a Retrieval-Augmented Generation (RAG) application using LangChain to reduce LLM hallucinations and enhance the information retrieval capabilities of language models with external knowledge sources.
- **Interdisciplinary Examples:** Creating an intelligent tutoring system for special education, developing an automated legal advisor for immigrants, building an interactive public policy FAQ generator.
- **Grading Rubric**
    - Integration and Functionality (30%): Effective integration of LangChain and GPT-3 for RAG.
    - Innovation and Relevance (25%): Originality and interdisciplinary relevance of the project.
    - Technical Proficiency (20%): Skillful use of RAG concepts and tools.
    - Code Quality (15%): Code efficiency, readability, and documentation.
    - Presentation and Report (10%): Coherence and thoroughness in project explanation.

Mini Project 4: Autonomous Multi-Agent Simulations

- **Libraries/Frameworks Used:** Python, AutoGen
- **Project Description:** Students will design, implement and analyze simulations of multi-agent systems to model complex social interaction scenarios between autonomous agents.
- **Interdisciplinary Examples:** Quantifying disinformation virality using multiagent simulations, modeling oxford style debates over social issues, creating economic analysis and consensus report from a committee of diverse rational agents.
- **Grading Rubric**
    - Simulation Complexity and Functionality (30%): Depth and functionality of the simulation.
    - Application to Real-World Problems (25%): Relevance to real-world interdisciplinary problems.
    - Agent Design and Interaction (20%): Complexity and effectiveness of agent behaviors and interactions.
    - Code Quality and Documentation (15%): Code structure, readability, and documentation.
    - Presentation and Analysis (10%): Effectiveness in presenting and analyzing simulation results.

Each project will be evaluated on these criteria, ensuring students not only develop technical skills but also appreciate the interdisciplinary applications of AI.

PROJECT ASSIGNMENTS:

Week 4 Start: (Mini Project #1) OpenAI API Function Calling App (covered Weeks 1-3)
Week 7 Start: (Mini Project #2) Huggingface Transformers Embeddings & Explainability Project (covered Weeks 4-6)
Week 10 Start: (Mini Project #3) LangChain RAG Project (covered Weeks 7-9)
Week 13 Start: (Mini Project #4) AutoGen Autonomous Multi-Agent Simulation Project (covered in Weeks 10-12)
Week 15 End: (Final Project) Original Interdisciplinary Research Project applying at least one key technology covered in 4 Mini-Projects

HOW THIS COURSE FITS INTO CURRICULUM:

Because the rapid pace of AI progress, many of these technologies have only appeared or become mature within the last year months with constant updates and refinements. This course differs from our survey of AI course (IPHS300 AI for the Humanities) which covers many more AI theories, models and social issues. In contrast, this course is a more focused exploration into 4 key leading edge AI technologies which we use to implement practical real-world research and industry solutions for fields like medicine, law or public policy.

SPECIAL TOPIC:

These are all very new AI technologies that have only recently matured and gained traction in both research and industry. A special topics course affords Kenyon to give students direct exposure to these cutting edge AI technologies which are invaluable in research and industry but evolving too quickly at this point to have been formalized into any fixed course or textbook. This also enables Kenyon to establish a core framework around rapid AI change in order to eventually develop a permanent course.