Large Language Model for Natural Language Processing

Lecturer: Sasha Apartsin

1 Background

Modern Natural Language Processing (NLP) is a field within artificial intelligence that focuses on enabling machines to understand, process, and generate human language in both written and spoken forms. It covers a broad range of tasks, including text classification, text summarization, text translation, question answering, and dialogue systems. With the rise of multimodal AI, which integrates language with other data types such as images, audio, and video, language has become a crucial component in tasks like text-to-image generation and image captioning.

2 Course Content

This course focuses on the implementation and application of Large Language Models (LLMs) for solving a wide range of natural language processing tasks. It covers the inner workings of LLMs and the creation of transformer-based foundation models through pretraining on large-scale datasets. Students will learn techniques for parameter-efficient fine-tuning (PEFT) to adapt these models to custom tasks using smaller, task-specific datasets. The course also explores techniques of LLM-based information retrieval, such as Retrieval-Augmented Generation (RAG), as well as models and tools for constructing agentic AI systems. Adopting a code-first approach, the course demonstrates key concepts and applications through practical coding examples using modern libraries and tools. These include the OpenAI API, LangChain, HuggingFace libraries, the DeepEval framework, and multi-agent development libraries such as LangGraph and CrewAI.

3 Course Project

During the course, students will propose, implement, and present an innovative project that builds on the concepts covered in class. The project will focus on a task of the student's choice, involving the generation of synthetic training data and the comparison of multiple LLMs, including both off-the-shelf and fine-tuned models. Each project team will give three in-class presentations: a project proposal, an interim progress report, and a final presentation. By the end of the course, teams will submit a GitHub repository containing all project materials, including presentation slides, source code, datasets, and a detailed README file. A sample of past student projects can be accessed through the following links: [link1], [link2]

4 Prerequisites

While all necessary background material, including machine/deep learning fundamentals and PyTorch library, will be introduced during the course, students are expected to have a solid understanding of foundational machine learning concepts and proficiency in Python. Prior experience with deep learning is recommended and will be beneficial.

5 AI TOOLS POLICY

The use of AI tools for generating project code and presentation slides is highly encouraged, provided that two essential requirements are met: the project must demonstrate novelty by addressing a new and valuable task, and the team must maintain full ownership and responsibility for all submitted code and presentation materials.

6 EXPECTED OUTCOMES

Graduates of the course will acquire a broad and in-depth understanding of techniques for constructing, training, and applying large language models. By the end of the course, students will have gained hands-on experience with state-of-the-art AI models and software libraries, developing practical skills through structured, guided projects. These projects will provide opportunities to design and implement AI solutions by combining, adapting, and extending existing components. The course emphasizes both technical proficiency and creative problem-solving, preparing students to innovate using modern AI technologies in real-world development scenarios. Students will develop a tangible AI project that can be showcased in a professional portfolio. At the same time, in-class presentations will enhance their ability to present and articulate complex technical work with confidence.

7 WEEKLY SCHEDULE

Below is an approximate weekly schedule outlining the subjects that will be covered. Please note that the actual order or content may vary depending on the class background and recent advancements in the field.

Week	Theme	Selected subjects
1	Course introduction	Course and project requirements Project Examples Basic AI concepts refresher
2	Natural Language Processing	Typical NLP tasks Cyber, healthcare, and software engineering use cases Tokenization, term and topic vectors, topic modelling Word embeddings
3	Introduction to Foundation LLM	LLM libraries and APIs Text representation and generation Text decoding, prompt engineering, and model fine-tuning
4	Synthetic data generation for model training and evaluation	Frameworks for synthetic data generation: DataDreamer, Curator Attributed and Bootstrapped data generation Weak supervision Data anonymization Model evaluation, text generation metrics, and LLM judges
5	Student presentations: Project Proposal	
6	Building and training transformer-based models	Attention, cross-attention, multiheaded attentions Transformer blocks, encoder-decoder transformers Transformer pretraining Sentence embedding Instruction following Fine-tuning for human preferences

		Seq2SeqModels Mixture of experts Music Transformer Explaining Transformers
7	LLM fine-tuning and transfer learning	PEFT: Adapters, Quantization, QLoRA Soft-prompts and p-tuning Representation fine-tuning: denoising Autoencoder, SimCSE Sequence and token classification fine-tuning: NER, SetFit Long texts and Longformer Model Distillation and merging
9	LLM for information retrieval and extraction	Vector stores: FAISS, Product Quantization, Chunking RAG: DPR, Chunking, HyPE, HyDE, MMR, RAG evaluation Topic Modelling and Recommendation Systems Knowledge Graphs: IE and KG-assisted generation
10	Student Presentations: Interim Report	
10	Agentic AI and LLM-based multi-agent systems	Tools: Function calls, MCP, ToolFormer Memory: MemGPT, and Conversational buffers Agent flows and planning: ReACT, Reflection, AgenticRAG, BabyAGI Multi-agent systems: AutoGen, CrewAI
11	Visual Language Models	Visual Transformer Multimodal models; CLIP/BLIP2 Document models: LayoutLM Segment Anything
12	Advanced topics	LLM Safety: LLM jailbreaking and watermarking LLM for Tabular Data: Text2SQL LLM for SE: requirements engineering, QA, design
13	Student Presentation: Project Finals	

8 COURSE GRADE

Delivery	Grade
In-class project proposal presentation	No grade, for feedback and approval only
In-class project interim presentation	20%
In-class final project presentation	40%
Final project submission (GitHub Repo)	40%

9 REFERENCES

1. Alammar, Jay, and Grootendorst, Maarten. Hands-On Large Language Models, 2024.
2. Raschka, Sebastian. Build a Large Language Model (From Scratch), 2024.
3. Tunstall, Lewis, et al. Natural Language Processing with Transformers, 2022.
4. Lanham, Michael. AI Agents in Action, 2025.
5. Kamath, Uday, et al. Large Language Models: A Deep Dive: Bridging Theory and Practice, 2024.