Course Title: Foundations of Large Language Models (LLMs)

Instructor: Ratnik Gandhi (ratnik.gandhi@villanova.edu) - **Office Hours:** Virtual by appointment
Teaching Assistant: Steve Halley (shalley@villanova.edu) - **Office Hours**: Tuesdays 4-6pm (Mendel 292)
Time and Classroom: Thursdays, 6:30pm-9:10pm in Mendel room G87

Course site: https://sites.google.com/view/foundations-of-llms

Target Audience: Advanced undergraduate or postgraduate students in computer science, data science, linguistics, or related fields.

Course Description: This course delves into the exciting world of large language models (LLMs), foundational models, and multimodal architectures, examining their theoretical underpinnings, training and fine-tuning methodologies, deployment strategies, and the challenges of hallucination, security, and reasoning. The landscape of the technology is rapidly changing and to accommodate most relevant new concepts, instructor may replace up to 20% of the course content.

Pre-requisite: Programming Language, Discrete Mathematics, Data Structures

Module 1: Introduction to LLMs (Week 1-3):

- A historical journey through the evolution of LLMs.
- Unveiling the mysteries of transformer architectures and attention mechanisms.
- Pre-training on massive datasets: delving into goals, challenges, and methodologies.
- Synthetic data for training LLMs.
- Exploring the diverse applications of LLMs across various domains.

Suggested Assignments:

- Setting up your LLM exploration environment with Hugging Face Transformers.
- Preprocessing text data for LLM training.

Module 2: Foundational Models and Multimodal Approaches (Week 4-6):

- Defining foundational models and their significance in the LLM landscape.
- Unveiling the power of multimodal models: integrating text, vision, and other data types.
- Multilingual LLMs
- A comparative analysis of LLMs, foundational models, and multimodal systems.
- Examining the limitations of model architectures, including the challenge of hallucination.

- Introducing chain-of-thought prompting for improved explainability and reasoning.
- Exploring the benefits of chain-of-thought for multimodal systems.
- RLHF for Multimodal Fusion and Explainability.

Suggested Assignments:

- Experimenting with a pre-trained multimodal LLM (e.g., VILT).
- Implementing chain-of-thought prompting for a simple multimodal task.

Module 3: Training and Fine-tuning Techniques (Week 7-9):

- Unveiling the pre-training objectives and algorithms used for unsupervised learning.
- Fine-tuning strategies for achieving exceptional task-specific performance.
- Prompt engineering: mastering the art of guiding LLM outputs.
- Exploring the potential of reinforcement learning and hyperparameter optimization.
- Hallucination risks in fine-tuning and specific techniques to mitigate them.
- A deep dive into chain-of-thought prompting as a fine-tuning technique.
- Understanding the methods for incorporating chain-of-thought into training, along with the challenges and limitations.
- RLHF and Proximal Policy Optimization (PPO) for Fine-tuning LLMs.

Suggested Assignments:

- Fine-tuning a pre-trained LLM on a specific NLP task.
- Implementing prompt engineering techniques to guide LLM outputs.
- Experimenting with chain-of-thought prompting for fine-tuning.
- Implementing RLHF using popular libraries like RLlib or Stable Baselines3

Module 4: Deployment and Productionalization (Week 10-12):

- Exploring deployment infrastructure options: cloud services (e.g., Google Cloud AI Platform, Amazon SageMaker), on-premise solutions (e.g., TPUs), and hybrid models.
- Model serving, optimization, and efficient inference techniques for real-time applications.
- Addressing bias, fairness, and ethical considerations when deploying LLMs.
- Monitoring, maintaining, and troubleshooting production LLMs for optimal performance.
- Highlighting the importance of hallucination monitoring and user-facing strategies in production.
- RLHF for Secure and Explainable LLM Deployment.

Suggested Assignments:

ratnik.gandhi@villanova.edu

- Deploying a trained LLM to a cloud platform (e.g., Google Cloud AI Platform).
- Implementing hallucination detection and mitigation techniques in production.
- Setting up monitoring and alerting systems for LLM performance.

Module 5: Future Directions and Research Frontiers (Week 13-14):

- Unveiling the latest research trends in LLMs: explainability, reasoning, and multimodality.
- Exploring the potential societal impacts, risks, and governance of LLMs.
- Delving into open research problems and the evolving landscape of LLM development.
- Introducing fully homomorphic encryption and its potential for secure LLM inference.
- Chain-of-thought prompting as a research direction for enhancing LLM reasoning.
- RLHF and the Future of LLM Reasoning and Explainability.

Suggested Assignments:

- Experimenting with chain-of-thought prompting for improved explainability and reasoning on LLMs.
- Brainstorming and designing a research project related to LLMs, addressing challenges like hallucination, reasoning, or security.

Resources:

- Blog: "OpenAI Blog" - Articles on responsible AI and LLM deployment https://openai.com/blog/: https://openai.com/blog/
- Paper: "The Troubling Emergence of Hallucination in Large Language Models" by Varshney https://arxiv.org/abs/2308.01778: https://arxiv.org/abs/2308.01778
- Online Course: "Neural Networks and Deep Learning" by deeplearning.ai on Coursera https://www.coursera.org/specializations/deep-learning: https://www.coursera.org/specializations/deep-learning
- Book: "Deep Learning" by Goodfellow, Bengio, and Courville (Chapter 10: Optimization) https://www.deeplearningbook.org/
- Paper: "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding" by Jacob Devlin et al. https://arxiv.org/abs/1810.04805
- Blog: "Hugging Face Blog" - Tutorials and articles on LLM training and fine-tuning https://huggingface.co/blog
- Paper: "Language Models are Few-Shot Learners" by Tom B. Brown et al. https://arxiv.org/abs/2005.14165
- Paper: "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions" by Dong et al. https://arxiv.org/abs/2311.05232

ratnik.gandhi@villanova.edu

- Paper: "Chain-of-Verification Reduces Hallucination in Large Language Models" by Li et al. https://arxiv.org/abs/2309.11495
- Online Course: "Natural Language Processing" by Stanford University on Coursera https://www.coursera.org/specializations/natural-language-processing
- Book: "Deep Learning" by Goodfellow, Bengio, and Courville (Chapter 16: Recurrent Neural Networks and LSTMs) https://www.deeplearningbook.org/
- Paper: "Attention Is All You Need" by Ashish Vaswani et al. https://arxiv.org/abs/1706.03762
- Blog: "The Gradient" - Articles on LLMs and NLP advancements https://thegradient.pub/
- Paper: "Chain-of-Thought Reduces Hallucination in Large Language Models" by Li et al. https://arxiv.org/abs/2309.11495: https://arxiv.org/abs/2309.11495
- Blog: "The Gradient" - Articles on cutting-edge LLM research https://thegradient.pub/: https://thegradient.pub/
- Articles and research papers on specific topics of interest (e.g., explainability, reasoning, secure inference)

Midterm Exam/Project: 10/10/2024 - Thursday from 6:15pm to 9:10pm
Final Project Presentation: 12/12/2024 - Thursday from 6:15pm to 9:10pm

Communication
For communication outside the classroom, email will be the main channel for this course. Therefore, it is crucial to check your email regularly. Emails will be sent to your Villanova email address. If we set up a MSTeams/Slack channel, details will be shared in the class.

Course Format
Each class typically features a focused lecture and/or hands-on activities, and sometimes debates or discussions. Students get collaborative guidance from the instructor and feedback from peers. Bringing a laptop is encouraged. There will be one or more short breaks during each session. Some assignments and/or project activities will be individual contributions and other would be collaborative group activities.

Grading Procedures
Your final grade will be calculated based on your performance across various areas, as outlined in the table below.

- Assignments: 30%
- Midterm Project/Exam: 20%
- Final Project: 40%
- Participation: 10%

ratnik.gandhi@villanova.edu

The instructor's grading approach offers many chances for students to show their grasp of the course material. The course emphasizes hands-on activities, including homework, in-class labs, and exercises. Further details are outlined below.

Homework Assignments
Weekly homework will be given, focusing on implementation techniques. These tasks will align with lecture topics, offering deeper exploration. Details will be posted on class website and discussed in class.

Assignments might include:

- Installing and configuring specific free software
- Writing software application code
- Creating scripts and similar artifacts
- Implementing configuration artifacts
- Setting up networking and connectivity
- Deploying applications locally or in the cloud
- Preparing for a class debate on a specific topic
- Researching and presenting findings on a special topic
- Creating and demonstrating a small program about a particular technology or
- technique

In-class Exercises
During the semester, students will engage in activities like programming labs, discussions, and debates. Their preparation, participation, and completion of these tasks will influence their grades. Given the limited class time, it's crucial to promptly advance through activities. Your instructor will assist, but don't hesitate to seek help if you're stuck.

Exams
There will be a midterm exam/project, and a final project presentation in this course. Since exams are less emphasized, these exams will cover material up to their dates as listed on the course calendar. The final project, encompassing content from the entire course, will be weighted more heavily and presented during last week of the course in class.

Before each exam, a review session will provide details about its format, question types, topics, and preparation tips.

Missed exams cannot be made up without prior instructor approval and must not be rescheduled for personal convenience. A valid, documented reason is required before the scheduled exam time. The instructor might assign a different and possibly harder exam if it's

ratnik.gandhi@villanova.edu

missed. In case of a genuine reason, like an emergency, students should contact the instructor as soon as possible using email.

Attendance Policy
You must attend all lectures, and attendance will be recorded at the start of each class. If you need to miss a class for a valid reason, inform the instructor promptly; otherwise, it will count as an unexcused absence and hurt your participation grade. If you miss any part of a class, you are responsible for catching up on missed work and materials and completing any in-class exercises. If the instructor must cancel a class due to their absence, students will be informed promptly. The instructor will collect students' contact details for such notifications.

Participation
A significant benefit of a course is the chance to delve into new concepts and have in-depth discussions with the instructor and classmates. Reflecting on ideas and methods, and sharing our thoughts openly in a respectful and non-judgmental setting, is key to the success of such courses. Therefore, it is essential that you attend class prepared and ready to discuss the assigned readings.

Active engagement in classroom exercises and discussions will boost your participation grade. Conversely, activities like browsing the internet, sending emails, or texting during class will negatively impact your participation grade. These distractions also disrupt your classmates' focus and the instructor's teaching. Thus, it's important to be courteous and attentive throughout each lecture.

Remember that your participation grade is based solely on your in-class interactions with students and the instructor. Please note that frequently emailing the instructor or meeting during office hours does not count towards improving your participation grade.

Late Projects and Assignments
Late assignments will face a substantial penalty. If you need more time, you must request an extension from the instructor before the assignment's due date. Additionally, you need to submit any work completed up to that point for your request to be considered. If an extension is approved, these grade penalties will apply:

- One day late: a 15% reduction.
- Two days late: a 25% reduction.
- Three days late: a 35% reduction.

Assignments submitted more than three days past the due date will not be accepted. Students are highly encouraged to meet deadlines. Adhering to strict deadlines is common in the software industry, so it's beneficial to get accustomed to them early in your career.

ratnik.gandhi@villanova.edu

Grading Scale

Your final letter grade will be based on a weighted average of all course components, rounded to the nearest whole number using standard rounding rules.

Final Average Final Grade

Final Average	Final Grade
93 – 100	A
90 – 92	A-
86 - 89	B+
83 – 85	B
80 – 82	B-
76 – 79	C+
73 – 75	C
70 – 72	C-
66 – 69	D+
63 - 65	D
60 - 62	D-
< 59	F

Development Environment

Considering the availability of open source libraries and development in area, this course will use Python as the primary programming languages. Further use of Git for version control and Google Colabs for experimentation and sharing assignments would be encouraged. Make sure to also familiarize yourself with a online storage drive for sharing assignments with your instructor and the TA.

Students with Disabilities

Students with physical or learning disabilities must register with the Learning Support Office to access appropriate academic accommodations for this course. Accommodations will not be granted without prior approval from the Learning Support Office. Additionally, please communicate with me after class or during office hours so I can understand your needs. The Learning Support Office can be contacted by phone at 610-519-5636, and their website is http://www.learningsupportservices.villanova.edu/. It is advisable to contact the Learning Support Office as early as possible, as the approval process for accommodation takes time.

Academic Integrity

Students need to understand the University's Academic Integrity Policy and the Dept. of Computing Sciences' policy, linked below. Complete all homework individually unless the instructor states otherwise in writing. If you have questions about ethics and integrity, ask your instructor.

http://www.vpaa.villanova.edu/academicintegrity/
http://csc.villanova.edu/academics/academicIntegrity

How to Use this Syllabus

Students need to thoroughly read the syllabus before the second class and ask any questions then. Periodically, they should revisit the syllabus to stay aware of the course overview and requirements.

How to Study for this Course

This course combines lectures and programming work, each requiring different strategies. Stay current with readings, actively engage with the material, and prepare discussion points for class. While lecture slides are provided, additional notetaking may be useful. Review notes weekly to prepare for exams and complete all homework to reinforce concepts.

Programming and technical classes have unique challenges. Feel free to ask the TA/instructor for assistance with study skills or any difficulties.

Sources and Credits

The instructor will use articles and resources from various online sources, generally acknowledging these excellent sources and making specific acknowledgements throughout the course as needed. E.g., Prof. Vega's DevOps Tools and Techniques syllabus has significantly contributed in preparing this document.

ratnik.gandhi@villanova.edu