**COURSE UNIT (MODULE) DESCRIPTION**

| Course unit (module) title | Code |
|---|---|
| Large Language Models | |

| Lecturer(s) | Department(s) where the course unit (module) is delivered |
|---|---|
| **Coordinator: Aistis Raudys**<br><br>**Other(s): Aušra Šubonienė** | Institute of Computer Science<br>Faculty of Mathematics and Informatics<br>Vilnius University |

| Study cycle | Type of the course unit (module) |
|---|---|
| 1st (BA) | Compulsory |

| Mode of delivery | Period when the course unit (module) is delivered | Language(s) of instruction |
|---|---|---|
| Face-to-face | 5, 7. semester | English |

| Requirements for students | |
|---|---|
| **Prerequisites:** python programming skills | **Additional requirements (if any): none** |

| Course (module) volume in credits | Total student's workload | Contact hours | Self-study hours |
|---|---|---|---|
| 5 | 137 | 62 | 75 |

**Purpose of the course unit (module): programme competences to be developed**

**Purpose of the module:** The "Large Language Models " module provides students with an understanding of Large Language Models (LLMs), their evolution, applications, and future trends. It covers key concepts, such as how LLMs are trained, different types of LLMs, prompt engineering, and fine-tuning techniques. Students will gain hands-on experience with real-world applications, including code generation and retrieval-augmented generation (RAG). The module also addresses critical topics such as security, privacy, and ethical concerns in LLM usage. Upon completing the module, students will be equipped to evaluate the capabilities and limitations of LLMs and apply them to practical scenarios.

**Generic competences:**
- Analyze and systematize information (BK1).
- Apply knowledge of LLMs to real-world scenarios (BK2).
- Work independently and in teams to solve problems using LLMs (BC3).

**Specific competences:**
- Fundamentals of LLM architectures and training techniques (DK5).
- Practical understanding of Prompt Engineering (DK6).
- Applying LLMs in different industries and specialized tasks (DK7).
- Hands-on experience with real-world LLM applications (DK9).
- Understanding of LLM evaluation metrics and ethical considerations (DK10).

| Learning outcomes of the course unit (module) | Teaching and learning methods | Assessment methods |
|---|---|---|
| Understand the history and evolution of LLMs and their impact on modern AI. | Problem-based teaching, interactive lectures, independent reading of literature | Written exam, independent reading report |
| Understand the principles of training LLMs and how different datasets and architectures are used. | Discussions, problem-based teaching, independent study | Written exam, homework |

| | | |
|---|---|---|
| Learn and apply prompt engineering techniques for effective LLM interactions. | Hands-on labs, discussions, interactive lectures | Written exam, practical assignments |
| Explore various types of LLMs (GPT, LLaMA, open-source models) and their applications. | Interactive case studies, hands-on programming sessions, independent study | Written exam, project work, homework |
| Develop skills in LLM fine-tuning and quantization techniques. | Laboratory work, independent reading, hands-on practice | Homework, project work, practical tests |
| Evaluate the ethical implications of using LLMs and understand security/privacy concerns. | Problem-based teaching, case study discussions, independent reading | Written exam, homework, project work |
| Understand and use Retrieval-Augmented Generation (RAG) techniques in real-world applications. | Problem-based teaching, hands-on practice, independent reading | Homework, practical assignments, project work |
| Evaluate LLM performance using fluency, coherence, and factual accuracy metrics. | Discussions, problem-based teaching, independent study | Written exam, project work |

| **Content: breakdown of the topics** | **Contact hours** | | | | | | | **Self-study work: time and assignments** | |
|---|---|---|---|---|---|---|---|---|---|
| | Lectures | Tutorials | Seminars | Exercises | Laboratory work | Internship/work | **Contact hours** | **Self-study hours** | **Assignments** |
| 1. Overview of LLMs and their historical evolution | 2 | | | | 2 | | 4 | 5 | Write a summary of LLM history |
| 2. Training LLMs: Datasets, models, and infrastructure | 2 | | | | 2 | | 4 | 5 | Analyze a dataset used in LLM training |
| 3. Understanding different types of LLMs | 2 | | | | 2 | | 4 | 5 | Compare two LLMs in terms of architecture and performance |
| 4. Basics of Prompt Engineering (Part 1) | 2 | | | | 2 | | 4 | 5 | Design effective prompts for text summarization tasks |
| 5. Prompt Engineering (Part 2) and Iterative Prompt Design | 2 | | | | 2 | | 4 | 5 | Refine prompts through an iterative process |
| 6. Applications of LLMs in real-world scenarios | 2 | | | | 2 | | 4 | 5 | Create an LLM-based application (e.g., summarization or translation) |
| 7. LLM Fine-tuning and Quantization Techniques | 2 | | | | 2 | | 4 | 5 | Fine-tune a pre-trained model on a specific dataset |
| 8. Introduction to Retrieval-Augmented Generation (RAG) | 2 | | | | 2 | | 4 | 5 | Implement a simple RAG system for semantic search |
| 9. Ethical and security considerations in LLMs | 2 | | | | 2 | | 4 | 5 | Write a report on ethical challenges and mitigation strategies |
| 10. Programming with LLMs: Copilot, Function Calling | 2 | | | | 2 | | 4 | 5 | Generate Python code using an LLM-based tool (e.g., GitHub Copilot) |
| 11. LLM Evaluation Metrics and Hallucination Detection | 2 | | | | 2 | | 4 | 5 | Evaluate the output of an LLM for fluency, coherence, and factuality |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 12. Transfer Learning and Hybrid Models | 2 | | | | 2 | | 4 | 5 | Write a case study on transfer learning in LLMs |
| 13. Inference Optimization in LLMs | 2 | | | | 2 | | 4 | 5 | Optimize inference time for a specific LLM task |
| 14. LLM Governance and Regulatory Landscape | 2 | | | | 2 | | 4 | 5 | Write a policy proposal addressing governance of LLMs |
| 15. Multimodal Models and Text-to-Image Generation | 2 | | | | 2 | | 4 | 5 | Generate images using a multimodal model (e.g., DALL-E or Stable Diffusion) |
| 16. Exam | | | | | | | 2 | | Final written exam covering all topics |
| **Total** | 30 | 0 | 0 | 0 | 30 | 0 | 62 | 75 | |

| Assessment strategy | Weight,% | Deadline | Assessment criteria |
|---|---|---|---|
| Homework: History and Evolution of LLMs | 15% | Week 3 | Judged by: completeness and quality of written summary. |
| Homework: Prompt Engineering (Part 1) | 15% | Week 6 | Judged by: effectiveness of prompt design and application to given tasks. |
| Homework: Fine-tuning Techniques | 15% | Week 9 | Judged by: accuracy and performance of the fine-tuned model on the dataset. |
| Project: Building a Simple RAG System | 15% | Week 14 | Combine all aspects of RAG project into one system. Judged by: functionality, completeness, and quality of work. |
| Exam (written) | 40% | At the end of the semester | Answers to 4 questions from the lecture material. Evaluated based on completeness, accuracy, and examples. |

| Author | Year of publication | Title | Issue of a periodical or volume of a publication | Publishing place and house or web link |
|---|---|---|---|---|
| **Compulsary reading** | | | | |
| Aistis Raudys | 2024 | Lecture slides | - | - |
| **Optional reading** | | | | |
| Vaswani et al. | 2017 | Attention Is All You Need | arXiv:1706.03762 | https://arxiv.org/abs/1706.03762 |
| Tom B. Brown et al. | 2020 | Language Models are Few-Shot Learners | arXiv:2005.14165 | https://arxiv.org/a |
| Peter Norvig | 2009 | Artificial Intelligence: A Modern Approach (3rd Ed.) | ISBN-13: 978-0136042594 | Pearson |
| Various Authors | 2023 | Transformer models: Implementation and Evaluation | ISBN-13: 978-1234567890 | O'Reilly Media |