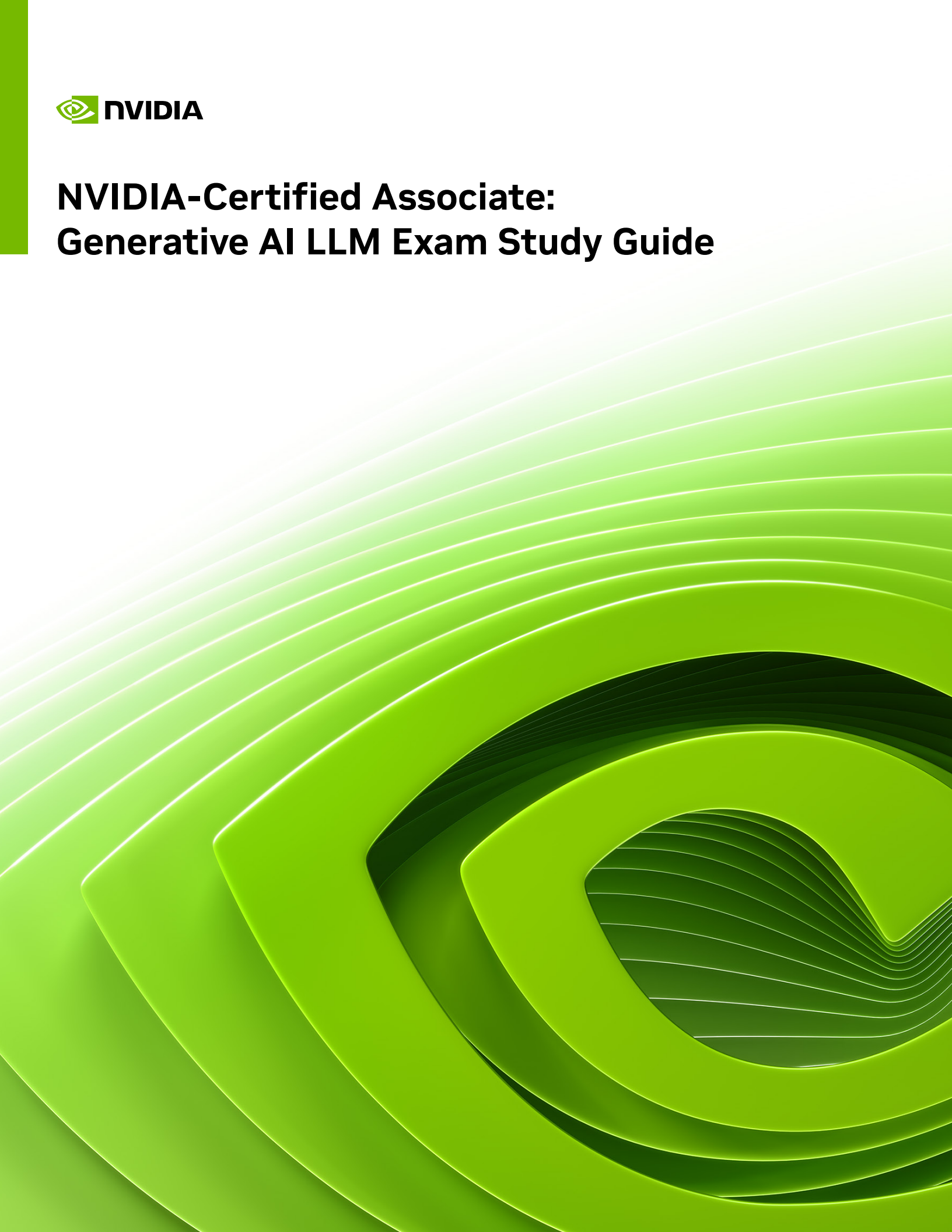
NVIDIA

NVIDIA-Certified Associate: Generative AI LLM Exam Study Guide



NVIDIA-Certified Associate: Generative AI LLM Exam Study Guide

Contents

This study guide provides an overview of each topic covered on the NVIDIA Generative AI LLM certification exam, recommended training, and suggested reading to prepare for the exam.

Information about NVIDIA certifications can be found **here**.

Job Description

The generative AI-large language model (LLM) associate developer is responsible for contributing to the development, programming, and quality assurance of state-of-the-art generative AI LLM systems. They work with a team of skilled AI professionals to develop datasets, select models to train, train models, and implement model testing and debugging processes. The associate should have an understanding of the deployment of models for applications. They'll also be responsible for developing high-quality software design and construction, programming in a variety of languages and platforms, and maintaining system updates.

Job Responsibilities

1. Collaborate with the AI development team to design, code, test, debug, and document programming applications.
2. Perform system analysis to ensure software and systems meet required specifications.
3. Aid in integrating new AI language models into existing systems or creating new ones as needed.
4. Assist in the assessment and resolution of application and system performance issues.
5. Stay updated on new AI models and other developments related to language learning models.
6. Contribute to the production of technical documents and manuals.
7. Conduct software programming and documentation development under the direction of senior staff.
8. Perform prompt engineering.
9. Assist in the process of model selection.
10. Define, curate, label, and annotate LLM datasets.
11. Experiment with A/B testing, evaluating prompts, evaluating models, and producing POCs.

Recommended Qualifications and Experience

1. Bachelor's degree in computer science, software engineering, AI, or a related field
2. Knowledge of Python, C, and AI frameworks (PyTorch, TensorFlow, etc.)
3. Solid understanding of neural networks and deep learning models

Certification Topics and References

Core Machine Learning and AI Knowledge: Exam Weight 30%

Knowledge of algorithms, conventions, and techniques that allow computers to learn from and make predictions or decisions based on data.

1.1	Assist in deployment and evaluation of model scalability, performance, and reliability under the supervision of senior team members.
1.2	Awareness of the process of extracting insights from large datasets using data mining, data visualization, and similar techniques.
1.3	Build LLM use cases such as retrieval-augmented generation (RAG), chatbots, and summarizers.
1.4	Curate and embed content datasets for RAGs.
1.5	Familiarity with the fundamentals of machine learning (e.g., feature engineering, model comparison, cross validation).
1.6	Familiarity with the capabilities of Python natural language packages (spaCy, NumPy, vector databases, etc.).
1.7	Read research papers (articles, conference papers, etc.) to identify emerging LLM trends and technologies.
1.8	Select and use models to create text embeddings.
1.9	Use prompt engineering principles to create prompts to achieve desired results.
1.10	Use Python packages (spaCy, NumPy, Keras, etc.) to implement specific traditional machine learning analyses.

Recommended Training (Optional)

Course reference: **Generative AI Explained**

- Define generative AI and explain how generative AI works.
- Describe various generative AI applications.
- Explain the challenges and opportunities of generative AI.

Course reference: **Prompt Engineering With LLaMA-2**

- Iteratively write precise prompts to bring LLM behavior in line with your intentions.

Suggested Readings

- **Attention Is All You Need**
- **End-to-End AI for NVIDIA-Based PCs: Transitioning AI Models With ONNX**, NVIDIA Technical Blog
- **Generative AI—What Is It and How Does it Work?**
- **Activation Function**
- **Implementing Deep Learning Methods and Feature Engineering for Text Data**
- **Autoregressive Model**
- **What Are Foundation Models?**, NVIDIA Blog
- **LoRA: Low-Rank Adaptation of Large Language Models**
- **Generative AI Research Spotlight: Demystifying Diffusion-Based Models**, NVIDIA Technical Blog
- **Training Hidden Units With Back Propagation**

Data Analysis: Exam Weight 14%

Inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, informing conclusions, and supporting decision-making.

2.1	Awareness of the process of extracting insights from large datasets using data mining, data visualization, and similar techniques.
2.2	Compare models using statistical performance metrics, such as loss functions or proportion of explained variance.
2.3	Conduct data analysis under the supervision of a senior team member.
2.4	Create graphs, charts, or other visualizations to convey the results of data analysis using specialized software.
2.5	Identify relationships and trends or any factors that could affect the results of research.

Recommended Training (Optional)

Course reference: **Introduction to Transformer-Based Natural Language Processing**

- Understand how transformer-based LLMs can be used to manipulate, analyze, and generate text-based data.

Course reference: **Fundamentals of Accelerated Data Science**

- Understand how to perform GPU-accelerated data manipulation.
- Ingest and prepare several datasets (some larger-than-memory) for use in multiple machine learning exercises.
- Read data directly to single and multiple GPUs with cuDF and Dask cuDF
- Prepare information for machine learning tasks on the GPU with cuDF.
- Apply several essential machine learning techniques to prepared data.
- Use supervised and unsupervised GPU-accelerated algorithms with cuML.
- Train XGBoost models with Dask on multiple GPUs.
- Create and analyze graph data on the GPU with cuGraph.
- Use NVIDIA RAPIDS™ to integrate multiple massive datasets and perform analysis.
- Implement GPU-accelerated data preparation and feature extraction using cuDF and Apache Arrow data frames.
- Apply a broad spectrum of GPU-accelerated machine learning tasks using XGBoost and a variety of cuML algorithms.
- Execute GPU-accelerated graph analysis with cuGraph, achieving massive-scale analytics in small amounts of time.
- Rapidly achieve massive-scale graph analytics using cuGraph routines.

Course reference: **Efficient Large Language Model (LLM) Customization**

- Assess the performance of fine-tuned models.

Suggested Readings

- **RAPIDS**
- **cuML 24.04.00 documentation**
- **GPU Accelerated Data Science With RAPIDS**
- **Data Exploration**
- **Stemming and Lemmatizing With sklearn Vectorizers**

Experimentation: Exam Weight 22%

The study of how to perform, evaluate, and interpret experiments, including AI model evaluation and the use of human subjects in labeling or reinforcement learning from human feedback (RLHF).

3.1	Awareness of the process of extracting insights from large datasets using data mining, data visualization, and similar techniques.
3.2	Compare models using statistical performance metrics, such as loss functions or proportion of explained variance.
3.3	Conduct data analysis under the supervision of a senior team member.
3.4	Create graphs, charts, or other visualizations to convey the results of data analysis using specialized software.
3.5	Identify relationships and trends or any factors that could affect the results of research.

Recommended Training (Optional)

Course reference: **Prompt Engineering With LLaMA-2**

- Leverage editing the powerful system message.
- Guide LLMs with one-to-many-shot prompt engineering.
- Incorporate prompt-response history into the LLM context to create chatbot behavior.

Course reference: **Fundamentals of Deep Learning**

- Enhance datasets through data augmentation to improve model accuracy.

Course reference: **Introduction to Transformer-Based Natural Language Processing**

- Understand how transformer-based LLMs can be used to manipulate, analyze, and generate text-based data.
- Leverage pretrained, modern LLMs to solve various natural language processing (NLP) tasks such as token classification, text classification.
- summarization, and question-answering.

Course reference: **Rapid Application Development With Large Language Models (LLMs)**

- Find, pull in, and experiment with the HuggingFace model repository and the associated transformers API.
- Use encoder models for tasks like semantic analysis, embedding, question-answering, and zero-shot classification.
- Use state management and composition techniques to guide LLMs for safe, effective, and accurate conversation.

Course reference: **Building RAG Agents for LLMs**

- Experiment with modern LangChain paradigms to develop dialog management and document-retrieval solutions.

Course reference: **Efficient Large Language Model (LLM) Customization**

- Use prompt engineering to improve the performance of pretrained LLMs.
- Apply various fine-tuning techniques.

Course reference: **Building Transformer-Based Natural Language Processing Applications**

- Leverage pretrained, modern LLM models to solve multiple NLP tasks such as text classification, named-entity recognition (NER), and question-answering.

Suggested Reading List

- **How to Conduct A/B Testing in Machine Learning?**
- **Inference Optimization**
- **Zero-Shot Testing**
- **Speech and Language Processing**
- **Machine Translation methods**
- **Hallucinations in Large Language Models**
- **General Language Understanding Evaluation**
- **Evaluating RAG Applications**
- **Cross-Validation in Machine Learning**
- **Benchmarking Elementary Language Tasks**
- **Building Transformer-Based Natural Language Processing Applications**

Software Development: Exam Weight 24%

Create, maintain, and test software.	
4.1	Assist in the deployment and evaluations of model scalability, performance, and reliability under the supervision of senior team member.
4.2	Build LLM use cases such as RAGs, chatbots, and summarizers.
4.3	Familiarity with the capabilities of Python natural language packages (spaCy, NumPy, vector databases, etc.).
4.4	Identify system data, hardware, or software components required to meet user needs.
4.5	Monitor functioning of data collection, experiments, and other software processes.
4.6	Use Python packages (spaCy, NumPy, Keras, etc.) to implement specific traditional machine learning analyses.
4.7	Write software components or scripts under the supervision of a senior team member.

Recommended Training (Optional)

Course reference: **Fundamentals of Deep Learning**

- Experience with common deep learning data types and model architectures.
- Leverage transfer learning between models to achieve efficient results with less data and computation.
- Take on your own project with a modern deep learning framework.

Course reference: **Introduction to Transformer-Based Natural Language Processing**

- Leverage pretrained, modern LLMs to solve various NLP tasks such as token classification, text classification, summarization, and question-answering.

Course reference: **Rapid Application Development With Large Language Models (LLMs)**

- Find, pull in, and experiment with the HuggingFace model repository and the associated transformers API.
- Use encoder models for tasks like semantic analysis, embedding, question-answering, and zero-shot classification.
- Use decoder models to generate sequences like code, unbounded answers, and conversations.
- Use state management and composition techniques to guide LLMs for safe, effective, and accurate conversation

Course reference: **Building RAG Agents for LLMs**

- Understand microservices, how to work between them, and how to develop your own.
- Practice with state-of-the-art models with clear next steps regarding productionalization and framework exploration.

Course reference: **Efficient Large Language Model (LLM) Customization**

- Leverage the NVIDIA NeMo™ framework to customize models like GPT, LLaMA-2, and Falcon with ease.

Course reference: **Building Transformer-Based Natural Language Processing Application**s

- Apply self-supervised transformer-based models to concrete NLP tasks using NVIDIA NeMo.
- Deploy an NLP project for live inference on NVIDIA Triton™.
- Manage inference challenges and deploy refined models for live applications.

Suggested Readings

- **TensorRT—Get Started**, NVIDIA Developer
- **Best Practices—NVIDIA NeMo**
- **Mastering LLM Techniques: Customization**, NVIDIA Technical Blog
- **Achieving FP32 Accuracy for INT8 Inference Using Quantization-Aware Training With NVIDIA TensorRT**
- **NCCL: Accelerated Multi-GPU Collective Communications**
- **Technologies Behind Distributed Deep Learning: AllReduce**, Preferred Networks Research & Development
- **Visual Intuition on Ring—Allreduce for Distributed Deep Learning**, by Edir Garcia Lazo, Towards Data Science
- **Big Data? Datasets to the Rescue!**, Hugging Face NLP Course
- **Deep Learning Scaling Is Predictable, Empirically**
- **BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding**

Trustworthy AI: Exam Weight 10%

Creation and assessment of ethical, energy-conscious, and reliable artificial intelligence systems capable of interpreting and integrating various forms of data, ensuring that they're designed and applied in a manner that's transparent, fair, and verifiable.

5.1	Describe the ethical principles of trustworthy AI.
5.2	Describe the balance between data privacy and the importance of data consent.
5.3	Describe how to use NVIDIA and other technologies to improve AI trustworthiness.
5.4	Describe how to minimize bias in AI systems.

Recommended Training (Optional)

Course reference: **Rapid Application Development With Large Language Models (LLMs)**

> Use state management and composition techniques to guide LLMs for safe, effective, and accurate conversation.

Course reference: **Generative AI With Diffusion Models**

> Learn about content authenticity and how to build trustworthy models.

Suggested Readings

> **Trustworthy AI for A Better World**, NVIDIA
> **What Is Trustworthy AI?**, NVIDIA Blog
> **What Is Retrieval-Augmented Generation aka RAG?**, NVIDIA Blogs

Questions?

Contact us **here**.

NVIDIA