

Course Conclusion

Congratulations! You've successfully completed the course and are now equipped with valuable insights into generative AI architecture and data preparation for large language models (LLMs). At this point, you know that:

- Generative AI refers to deep-learning models that can generate content, such as text, images, audio, 3D objects, and music, based on the training data.
- These models understand the relationship between words and phrases and generate contextually relevant text. An example of a generative AI model for text generation is a generative pre-trained transformer (GPT).
- These models can generate images from text input and seed images. Examples of generative AI models for image generation are data analysis learning with language model for generation and exploration (DALL-E) and generative adversarial networks (GANs).
- Generative AI models can generate natural-sounding speech. An example of this type of generative AI model is WaveNet.
- Generative AI architectures and models include recurrent neural networks (RNNs), transformers, GANs, variational autoencoders (VAEs), and diffusion models.
 - RNNs use sequential or time series data and a loop-based design for training.
 - Transformers utilize the self-attention mechanism to focus on the most important parts of the information.
 - GANs consist of a generator and a discriminator, which work in a competitive mode.
 - VAEs operate on an encoder-decoder framework and create samples based on similar characteristics.
 - Diffusion models generate creative images by learning to remove noise and reconstruct distorted examples, relying on statistical properties.
- Generative AI started with rule-based systems that use predefined linguistic rules, followed by machine-learning approaches focusing on statistical methods. It later moved to deep learning, which uses neural networks trained on extensive data sets. Transformers represent the latest in this evolution.

- The evolution of generative AI has led to advancements in machine translation, chatbot conversations, sentiment analysis, and text summarization.
- Large language models (LLMs) are foundation models that use AI and deep learning with vast data sets. LLMs have training data sets that run into petabytes and contain billions of parameters. Examples of LLMs are GPT, Bidirectional Encoder Representations from Transformers (BERT), Bidirectional and Auto-Regressive Transformers (BART), and Text-to-Text Transfer Transformer (T5).
- Some libraries and tools you can use to implement generative AI for NLP are PyTorch, TensorFlow, Hugging Face, LangChain, and Pydantic.
- Tokenization and data loading are part of the data preparation activities for natural language processing (NLP).
- Tokenization breaks a sentence into smaller pieces or tokens. These tokens can be words, characters, or subwords, making complex text understandable to computers. Examples of tokenizers are natural language toolkit (NLTK) and spaCy.
- Word-based tokenization preserves the semantic meaning, though it increases the model's overall vocabulary.
- Character-based tokenization has smaller vocabularies but may not convey the same information as entire words.
- Subword-based tokenization allows frequently used words to stay unsplit while breaking down infrequent words.
 - Using the WordPiece, Unigram, and SentencePiece algorithms, you can implement subword-based tokenization.
- You can add special tokens such as <bos> at the beginning and <eos> at the end of a tokenized sentence.
- A data set in PyTorch is an object that represents a collection of data samples. Each data sample typically consists of one or more input features and their corresponding target labels.
- A data loader helps you prepare and load data to train generative AI models. Using data loaders, you can output data in batches instead of one sample at a time.
- Data loaders seamlessly integrate with the PyTorch training pipeline and simplify data augmentation and preprocessing.