Summary and Highlights

Congratulations! You have completed this lesson. At this point in the course, you know that:

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.

Tokenizers are essential tools that break down text into 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 have several key parameters, including the data set to load from, batch size (determining how many samples per batch), shuffle (whether to shuffle the data for each epoch), and more. Data loaders also provide an iterator interface, making it easy to iterate over batches of data during training.

PyTorch has a dedicated DataLoader class.

Data loaders seamlessly integrate with the PyTorch training pipeline and simplify data augmentation and preprocessing.

A collate function is employed in the context of data loading and batching in machine learning, particularly when dealing with variable-length data, such as sequences (e.g., text, time series, and sequences of events). Its primary purpose is to prepare and format individual data samples (examples) into batches that machine learning models can efficiently process.