Advanced Generative AI: Models, Tools and Applications

Large Language Models

Quick Recap

- How do Generative AI features contribute to different domains like healthcare, finance, and others?
- What emerging trends in Generative AI do you foresee shaping the future?

Engage and Think

What if Large Language Models (LLMs) could generate completely original and human-like text in any language or programming language?

How would this revolutionize the way humans communicate and interact with technology?

Learning Objectives

By the end of this lesson, you will be able to:

- Develop an understanding of the core components and architecture of Large Language Models (LLMs)
- Experiment with analyzing the LLM in action and its training process, encompassing tokenization, embedding, neural network training, and fine-tuning
- Identify the functioning of LLMs, focusing on how they generate human-like text and respond to prompts
- Organize a comparison and contrast of various LLMs

Language Models

Language Models

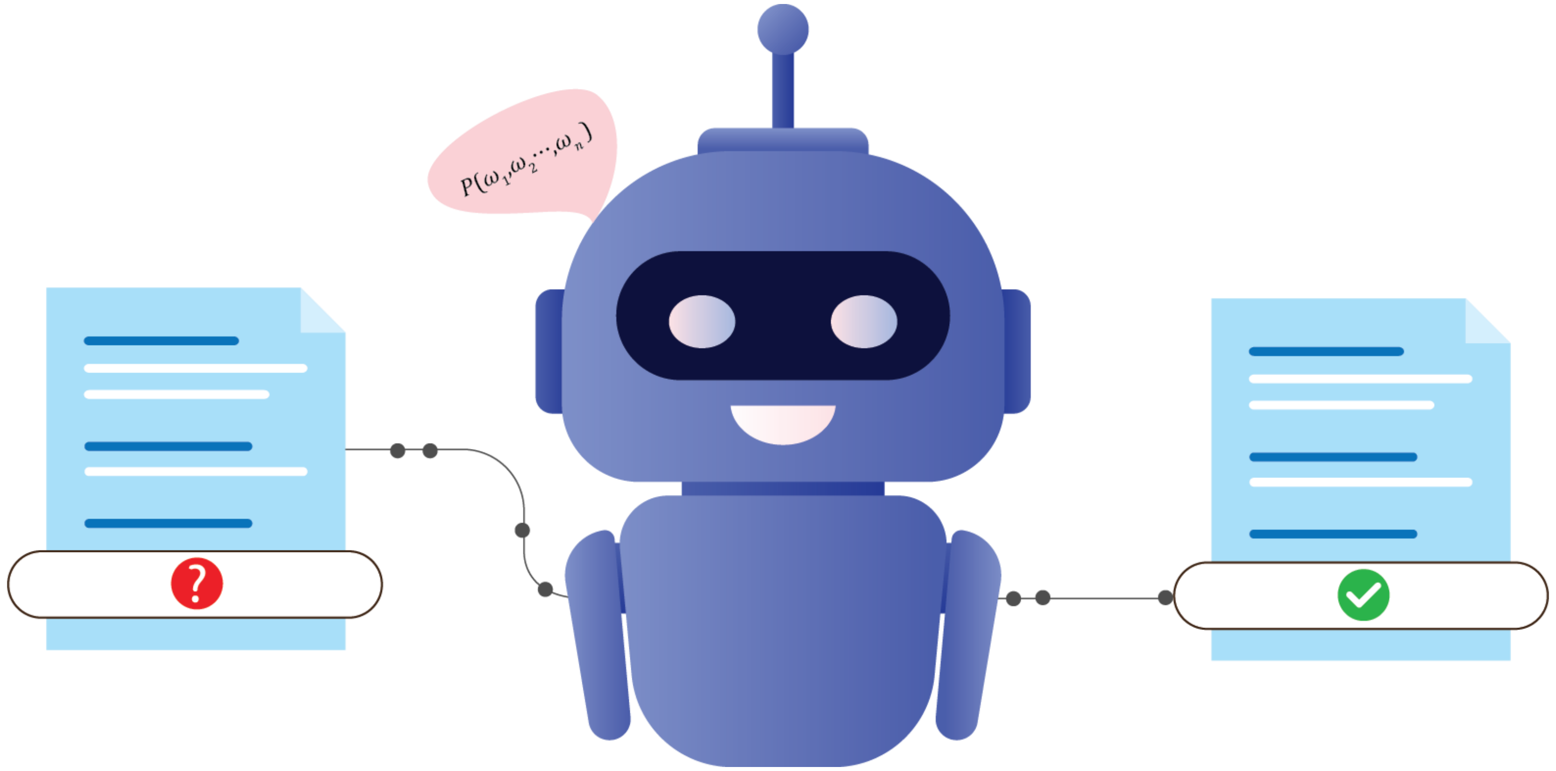
A language model is a probabilistic machine learning entity.

It resembles a complex function, designed to predict the probability of word sequences within a specific language corpus.

It is represented as: P(**Any sentence here**)

Language Models: Equation

Language models operate by assigning probabilities to sequences of words.



Mathematically, it looks like this:

$$P(\omega_1, \omega_2 \cdots, \omega_n) = P(\omega_1) \cdot P(\omega_2|\omega_1) \cdot P(\omega_3|\omega_1, \omega_2) \cdot \ldots \cdot P(\omega_n|\omega_1, \omega_2, \ldots, \omega_{n-1})$$

Language Models: Example

Consider the sentence: This is a new technology.

The language model calculates the probability of the sentence as:

P**(This is a new technology)**

$P(\text{This is a new technology}) = P(\text{This})\ P(\text{is}|\text{This})\ P(\text{a}|\text{This is})\ P(\text{new}|\text{This is a})\ P(\text{technology}|\text{This is a new})$

Language Models: Calculation

To illustrate, let's calculate the probability of two different sentences:

1. P(This is a fluffy dog.)
2. P(This are a purple flying deer.)

Solution: Sentence 1 gets a high probability, leveraging common context, and in sentence 2, rare and challenging words result in a lower probability.

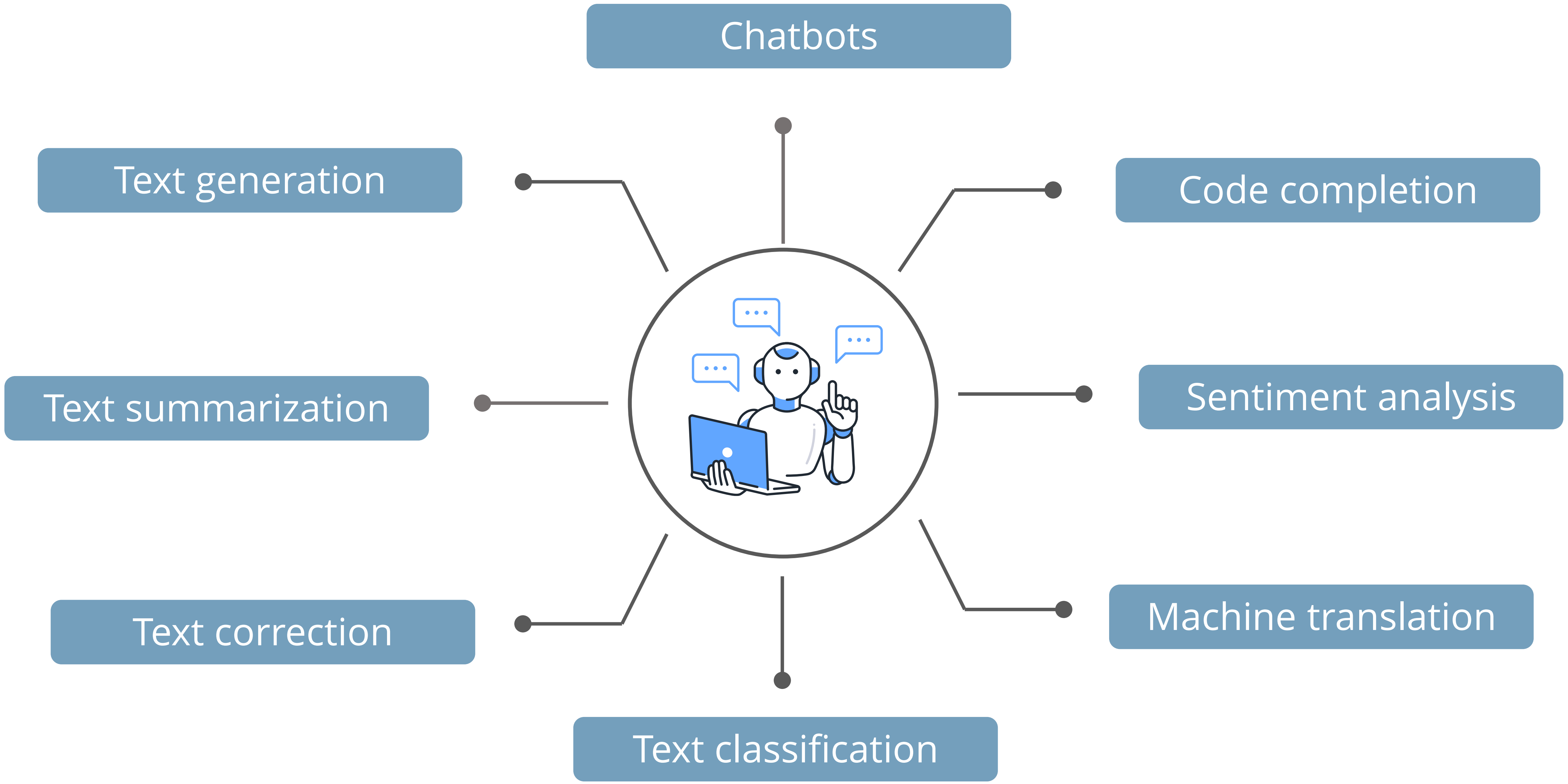
Power of Language Models

The powers of language models extend beyond just sentence prediction.

They are incredibly versatile.

They can answer questions.

Applications of Language Models



Demo: Text Generation

Duration: 20 minutes

Imagine you are on a quest to understand the intricate art of text generation, where a computer learns the patterns of a given writing style and crafts its sentences.

Today's session will explore a Python script designed for educational purposes. This script employs the Natural Language Toolkit (NLTK) and the Brown corpus to demonstrate text generation through a Markov chain model using trigrams.

Note

Please download the solution document from the Reference Material Section and follow the Jupyter Notebook for step-by-step execution.

DEMONSTRATION

Quick Check

Which of the following is not an application of language models?

A. Text generation

B. Machine translation

C. Speech recognition

D. Image processing

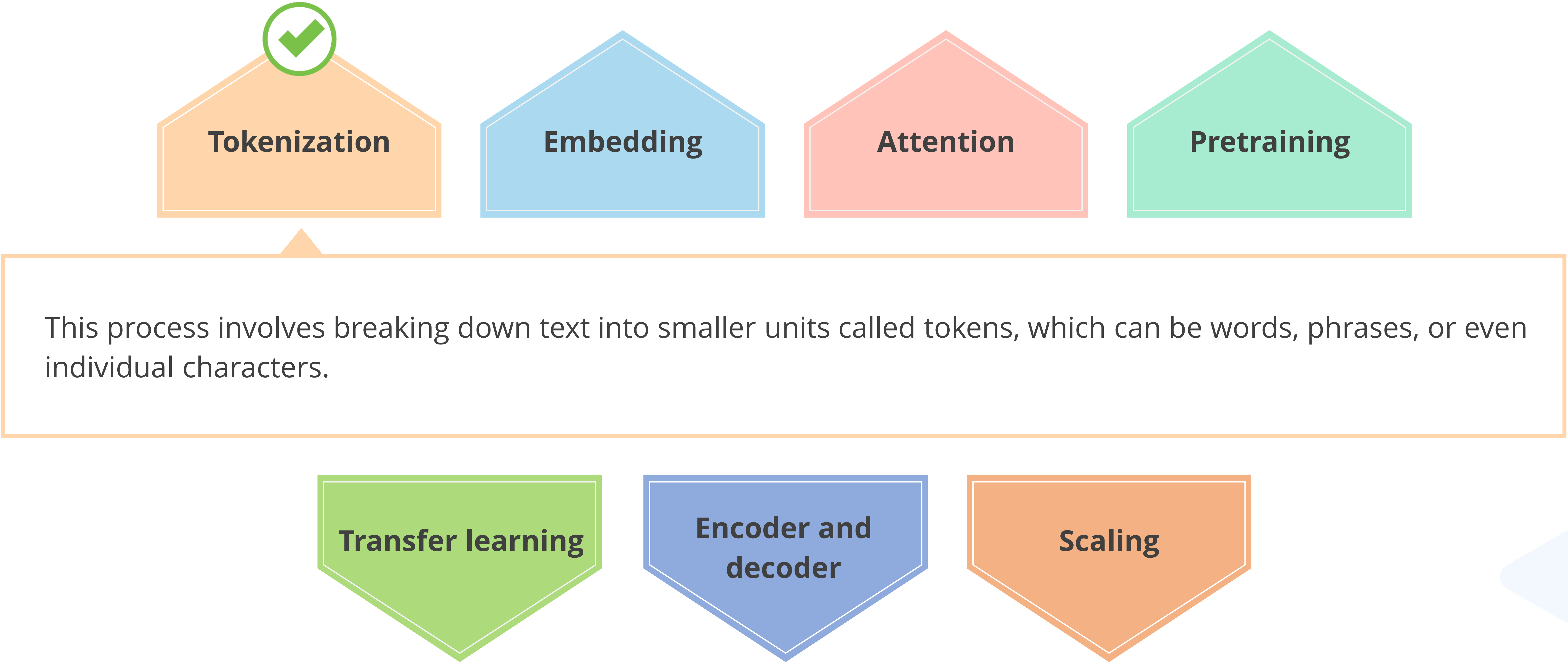
Large Language Models

Large Language Models

Large Language Models (LLMs) are state-of-the-art AI models designed to comprehend and generate human language.

Large	Refers to the significant size and complexity of these models, which contains hundreds of millions or even billions of parameters
Language	Denotes their primary function, which is to understand and generate human language
Model	Describes them as mathematical representations that capture the patterns and structure of language data

Components of LLMs



Components of LLMs

- Tokenization
- Embedding
- Attention
- Pretraining

This embedding component maps tokens to a high-dimensional vector space, representing each token with a unique vector.

- Transfer learning
- Encoder and decoder
- Scaling

Components of LLMs

Tokenization

Embedding

Attention

Pretraining

This attention mechanism lets the model concentrate on specific parts of the input text when generating output.

Transfer learning

Encoder and decoder

Scaling

Components of LLMs

- Tokenization
- Embedding
- Attention
- Pretraining

This involves pretraining LLMs on extensive text data to understand the underlying patterns and structures of human language.

- Transfer learning
- Encoder and decoder
- Scaling

Components of LLMs

Tokenization

Embedding

Attention

Pretraining

This component allows the model to adapt to new tasks by fine-tuning the pre-trained model on a smaller dataset.

Transfer learning

Encoder and decoder

Scaling

Components of LLMs

- Tokenization
- Embedding
- Attention
- Pretraining

This employs the Transformer framework in a large language model architecture, comprising two main parts: an encoder and a decoder.

- Transfer learning
- Encoder and decoder
- Scaling

Components of LLMs

Tokenization

Embedding

Attention

Pretraining

This necessitates significant computational resources for training and upkeep, making scaling a challenging but essential part of its architecture.

Transfer learning

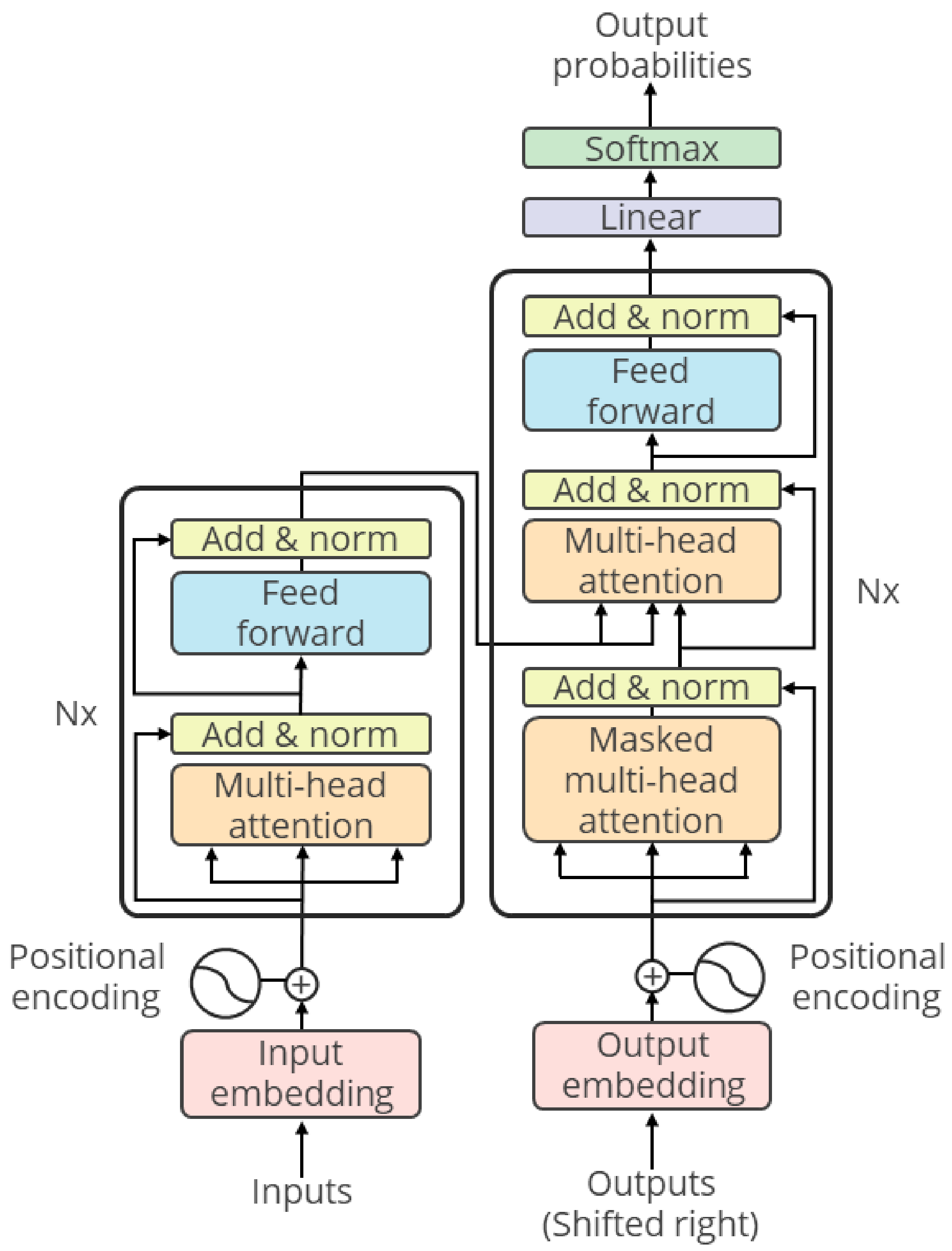
Encoder and decoder

Scaling

LLM Architecture

Components of LLM architecture

- Input embeddings
- Positional encoding
- Encoder
  - Attention mechanism
  - Feed-forward neural network
- Decoder
- Multi-headed attention
- Layer normalization
- Output



LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- Output

- The machine takes in a sentence and breaks it down into smaller pieces.
- Each of these pieces is turned into a special kind of code that the machine can understand.
- This code holds the meaning of the words.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- Output

- The machine wants to understand not just what words are there but also their order in the sentence.
- So, it adds some extra information to the code to show where each word is in the sentence.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- Output

- **Encoder:** Now, the machine gets to work on analyzing the sentence. It creates a bunch of memories to remember what it has read.
- **Attention mechanism:** The machine pays more attention to some words depending on their importance in the sentence.
- **Feed forward:** After paying attention to words, the machine thinks hard about each word on its own.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- **Decoder**
- Multi-headed attention
- Layer normalization
- Output

- The machine not only understands but also generates new sentences.
- For this, it has a special part called the decoder.
- The decoder helps the machine predict what word comes next based on what it has understood so far.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- Output

- The machine looks at the words in different ways simultaneously.
- This helps the machine grasp different aspects of the sentence all at once.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- Output

- This layer is in place to keep everything in check and make sure the machine learns well.
- The machine normalizes its understanding at each step.

LLM Operations

These represent the functions of components within an architecture.

- Input embeddings
- Positional encoding
- Encoder
- Decoder
- Multi-headed attention
- Layer normalization
- **Output**

- Finally, the machine produces its own understanding or generates new sentences.
- The output depends on what the machine is designed to do.
- For example, if it's predicting the next word in a sentence, it gives a probability for each word.

LLM Training Steps

The steps in the training process of a language model are:

Corpus preparation

Tokenization

Neural network training

Embedding generation

Neural Network Training

It empowers advanced neural network architectures to train the LLM on the curated dataset, fine-tuning parameters to enhance its capacity for generating coherent and contextually accurate language.

Once the neural network is trained, further fine-tuning is often conducted using techniques like Reinforcement Learning with Human Feedback (RLHF) to align the model's outputs with specific goals, such as generating user-preferred responses or avoiding biased content.

Reinforcement Learning with Human Feedback (RLHF)

It is a fine-tuning technique that incorporates human preferences to improve the performance and alignment of large language models. It involves:

Collecting feedback	Human evaluators provide feedback on model-generated outputs, ranking them based on quality or relevance.
Reward model training	A reward model is trained using the feedback to guide the LLM toward preferred behavior.
Policy optimization	he LLM is fine-tuned using reinforcement learning techniques, such as proximal policy optimization (PPO), to maximize the reward signal.

This process enhances the model's ability to generate more contextually accurate, aligned, and user-centric outputs, making it an essential step in modern LLM training pipelines.

Quick Check

When considering the architecture of Large Language Models (LLMs), which of the following components is responsible for generating human-like text and responding to prompts?

A. Tokenization

B. Embedding

C. Neural network training

D. Fine-tuning

Types of Large Language Models (LLMs)

Types of LLMs

Below are the various pretrained LLMs available in the market:

GPT 3.5
and GPT 4

PaLM

Claude

Cohere

Falcon

LLaMA

Types of LLMs: GPT 3.5

This model is a sophisticated addition to OpenAI's GPT series, pushing the boundaries of language processing.

Performance

It delivers outstanding performance across a variety of natural language processing tasks.

Pros

The model excels at executing a broad spectrum of natural language processing tasks.

Cons

Compared to GPT-4, this model may generate more restricted content and is considered less advanced.

Types of LLMs: GPT 4

This is a big language model created by OpenAI. It uses GPT-3's strengths, reaching new levels of scale and performance.

Performance	It performs like a human on tests, scoring in the top 10% on a simulated bar exam.
Pros	This is a top-notch language model, handling tough problems more accurately and is multimodal.
Cons	It is likely to be more expensive than other language models.

Types of LLMs: PaLM 2

This is a Google AI-developed next-gen LLM.

Performance	It excels in reasoning tests, outdoing its predecessor on various NLP benchmarks.
Pros	It can process both text and image inputs.
Cons	It's not as commonly used as other models and may lack extensive support.

Types of LLMs: Claude V1

This is a big language model crafted by Anthropic, an AI research company.

Performance	It performs better than many earlier language models, such as GPT-3 and older OpenAI models, in generating longer, detailed responses. It can be accessed via an API and the public beta site, claude.ai.
Pros	It creates clear and interesting answers, and you can fine-tune it for specific topics.
Cons	It requires a large amount of training data to achieve optimal performance.

Types of LLMs: Cohere

This is a big language model made by Cohere Technologies.

Performance	It excels in various natural language processing tasks, showing remarkable performance.
Pros	It manages various tasks and can be fine-tuned for specific areas.
Cons	Training it demands a lot of computational resources.

Types of LLMs: Falcon

This is a foundational large language model from the Technology Innovation Institute (TII) in the United Arab Emirates.

Performance	Its performance stands out, boasting high accuracy, robustness, and efficiency.
Pros	It is known for its quick processing speed, which makes it perfect for real-time applications.
Cons	It might not be suitable for tasks requiring advanced natural language processing capabilities.

Types of LLMs: LLaMA

This is a family of LLMs launched by Meta AI in February 2023.

Performance	It shows outstanding performance in various natural language processing tasks.
Pros	It understands context-rich information, enhancing its effectiveness in complex tasks.
Cons	It might unintentionally produce biased or inaccurate content.

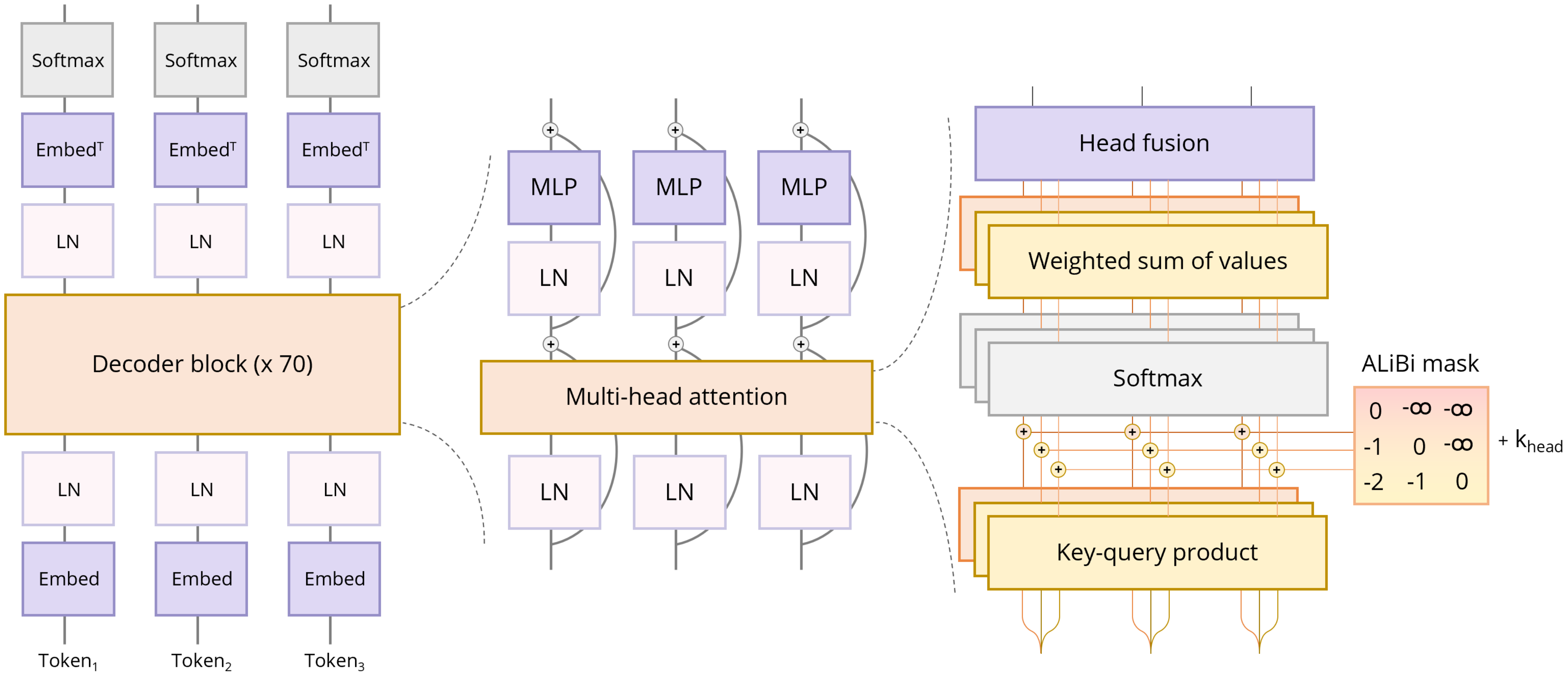
Bloom

Bloom Overview

It is an autoregressive Large Language Model trained on extensive text data using industrial-scale computational resources.

Bloom's Architecture

BLOOM adopts a conventional decoder-only transformer architecture.



Bloom’s Architecture

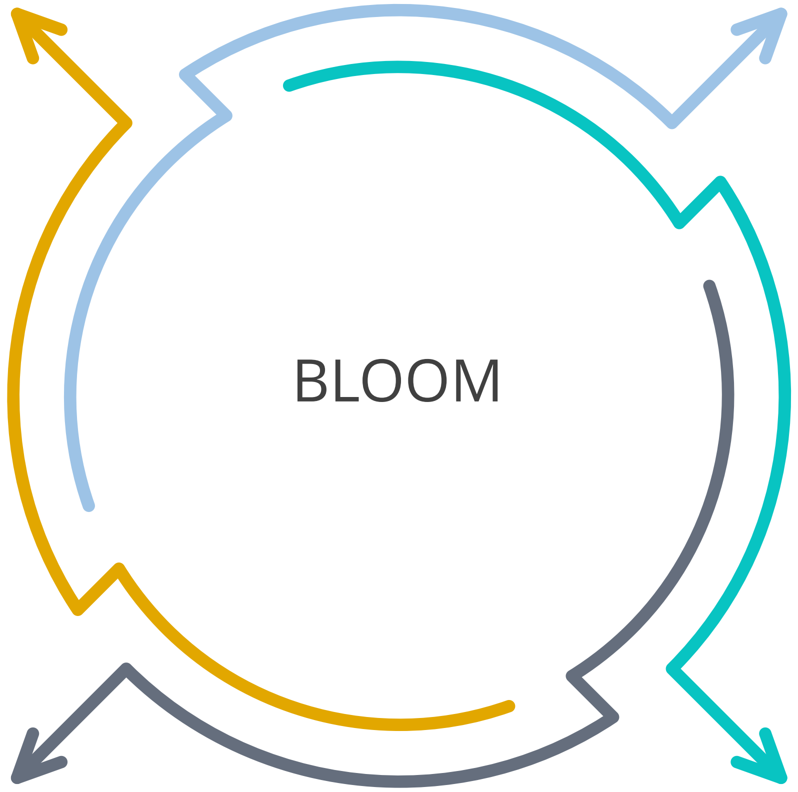
It features several notable modifications, including:

ALiBi	This component enhances the model's capacity to generalize to longer context lengths beyond what it encounters during training.
Embedding layer norm	An additional layer of normalization is introduced after the model's embedding layer, contributing to enhanced training stability.

Unpacking Bloom

It is trained on a massive 1.6TB of text data.

It boasts a staggering 176 billion parameters.



It excels in text generation: 46 natural and 13 programming languages.

It is an architecture, rooted in an auto-regressive model.

LLM Reasoning

Diverse reasoning	The LLM explores varied reasoning, including common sense and math, adapting to diverse contexts.
Eliciting reasoning	Methods like chain-of-thought prompting guide LLMs to stimulate and prompt thoughtful reasoning.
Reasoning contribution enigma	The challenge lies in understanding reasoning's role and impact, differentiating it from factual information.

Quick Check

Which method can be utilized to unleash the reasoning capabilities of LLMs?

A. Cross-Modal Learning

B. Few-Shot Learning

C. Chain-of-Thought Prompting

D. Self-Supervised Learning

LLM Considerations

LLM Considerations

There are two types of considerations for choosing an LLM:

Critical considerations:

Evaluate non-technical aspects, like ethics and biases.

Technical considerations:

Assess performance, architecture, and computational requirements.

Critical Considerations

The critical considerations for choosing an LLM are:

- Licensing and commercial use
- Practical factors for inference speed and precision
- The impact of context length and model size
- Task-specific vs. general-purpose
- Testing and evaluation
- Deployment cost considerations

Technical Considerations

The technical considerations for choosing an LLM are:

- Data security and privacy
- Model inference monitoring
- Scalability and performance
- Version control and updating
- APIs and integration security

Guided Practice

Overview **Duration: 25 minutes**

This activity focuses on testing understanding of diverse language models and their applications. It presents scenarios that require applying learned concepts to solve problems or accomplish tasks.

GUIDED PRACTICE

Key Takeaways

- Language model is a machine learning entity.
- Large Language Models are trained on large datasets, and they can generate human-like text, images, and many more.
- Pretrained LLMs available in the market can be utilized for powerful generative AI solutions
- Bloom is an autoregressive LLM capable of generating text in 46 natural languages and 13 programming languages.

Q&A