DESCRIPTIVE QUESTIONS-

Q1. What are Generative Models?  
Ans: Generative models, as the name suggests, are designed to generate new data instances that closely resemble those found in a given training dataset. These models capture the underlying patterns and structure of the training data and use that knowledge to produce new samples. The primary purpose of generative models is to generate data that is statistically similar to the original dataset, with the goal of creating samples that are indistinguishable from real data instances.

Q2. What are the types of Generative Model?  
Ans: **Types of Generative Models**

* + Probabilistic Generative Models: Model the probability distribution of the data.
  + Autoencoders: Learn a compact representation of the input data.
  + Variational Autoencoders (VAEs): Combine aspects of autoencoders with probabilistic generative models.
  + Generative Adversarial Networks (GANs): Learn a generator to produce data and a discriminator to distinguish between real and generated samples.

Q3. What is Language Modeling (LM)?  
Ans: Language Modeling (LM)-

Definition: Language modeling is a task in natural language processing (NLP) where the goal is to predict the probability of a sequence of words occurring in a given context.

Applications: Speech recognition, machine translation, text completion.

Q4. What are some challenges in Language Modeling (LM)?  
Ans: Some challenges in Language Modeling are as follows-

Data Sparsity: Language models need to capture the diversity and variability of natural language, which can be challenging with limited training data. This problem is particularly acute for less common words or phrases, leading to data sparsity issues.

Out-of-Vocabulary Words: Handling out-of-vocabulary (OOV) words, which are words not seen during training, is crucial for effective language modeling. Coping with unseen words while maintaining coherence and relevance in generated text is a significant challenge.

Long-Term Dependencies: Capturing long-term dependencies in language is essential for generating coherent and contextually relevant text. Traditional recurrent neural networks (RNNs) suffer from vanishing gradient problems, making it difficult to model long-range dependencies effectively.

Contextual Ambiguity: Natural language is inherently ambiguous, with words often having multiple meanings depending on the context. Resolving such ambiguity requires models to incorporate rich contextual information, which can be challenging, especially in complex sentences.

Evaluation Metrics: Evaluating the performance of language models is non-trivial. Traditional metrics such as perplexity provide an indication of how well a model predicts the next word but may not always correlate with human judgment regarding the quality or fluency of generated text.

Q5. What are some examples of LLM?  
Ans: **GPT-3 (Generative Pre-trained Transformer 3):** Developed by OpenAI, this is one of the most prominent LLMs, producing coherent, contextually appropriate text. It’s already being widely used in applications including chatbots, content generation, and language translation.

**GPT-4:** This successor to GPT-3 supplies advancements in contextual understanding and memory capabilities. As an evolving model, the goal is to further improve the quality of generated text and push the boundaries of language generation.

**PaLM 2 (Pre-trained AutoRegressive Language Model 2):** Here’s a non-GPT example of an LLM that’s focused on language understanding and generation, offering enhanced performance in tasks such as language modeling, text completion, and document classification. With this functionality, it does a good job of powering the Google Bard chatbot.

**Generative AI plus LLMs:** a dynamic duo

Now that you have an idea of how generative AI and large language model technology works in some real-world areas, here’s something else to think about: when they’re utilized together, they can enhance various applications and unlock some exciting possibilities.

Q6. What is self-attention mechanism?  
Ans: The self-attention mechanism, also known as intra-attention, is a key component in many modern neural network architectures, particularly in the field of natural language processing (NLP). It enables the network to weigh the importance of different words in a sentence or sequence when processing that sequence. The mechanism allows the model to focus on relevant parts of the input while performing tasks such as language modeling, machine translation, and text classification.  
A single self-attention mechanism is called a head. The head works as follows. First, the input is fed into three separate linear layers. Two of those (the queries Q and the keys K) are multiplied, scaled, and turned into a probability distribution using a softmax activation function. Think of this probability distribution as describing which indices matter most for the output (i.e. which words in the prompt matter for the next word to be predicted). Finally, the output is multiplied with values V. This thus gives V \* the importance of each of the tokens in V. A key observation is that the learnable parameters in the head are the three linear layers.

Q7. What is multi head attention?  
Ans: Multi-head attention is nothing more than several individual heads stacked on top of one another. The input to all heads is equivalent. However, each head has its own weights. After forwarding the input through all the heads, the output of the heads is concatenated and passed through a linear layer which brings the dimensionality back to the dimension of the initial input.

Q8. What is Encoder-Only models?  
Ans: Encoder models use only the encoder of a Transformer model. At each stage, the attention layers can access all the words in the initial sentence. These models are often characterized as having “bi-directional” attention, and are often called auto-encoding models.

The pretraining of these models usually revolves around somehow corrupting a given sentence (for instance, by masking random words in it) and tasking the model with finding or reconstructing the initial sentence.

Encoder models are best suited for tasks requiring an understanding of the full sentence, such as sentence classification, named entity recognition (and more generally word classification), and extractive question answering.

Q9. What is BERT?  
Ans: ***BERT (Bidirectional Encoder Representations from Transformers)***

BERT is one of the most prominent pre-trained transformer models developed by Google. It’s bidirectional, meaning it processes words concerning all the other words in a sentence, rather than one by one in order. The following are some of the unique features of BERT model:

Q10. Write down some key points about BERT.  
Ans: Here are some key points about BERT:

1. Bidirectional: BERT is bidirectional, meaning it can understand the meaning of a word in a sentence by considering both the left and right context surrounding that word. This bidirectional context modeling is achieved using the self-attention mechanism within the transformer architecture.

2. Pre-training: BERT is pre-trained on large corpora of text data using two unsupervised learning tasks: masked language modeling (MLM) and next sentence prediction (NSP). During MLM, a certain percentage of the input tokens are randomly masked, and the model is trained to predict the original tokens based on the context. NSP involves predicting whether a given sentence follows another sentence in the input text.

3. Transformer Architecture: BERT is based on the transformer architecture introduced by Vaswani et al. It consists of multiple layers of self-attention and feedforward neural networks. This architecture allows BERT to capture long-range dependencies in text data efficiently.

4. Contextual Embeddings: BERT produces contextual word embeddings, which means that the representation of a word can vary depending on its context within a sentence. This allows BERT to capture nuances in meaning and resolve ambiguities more effectively compared to traditional word embeddings like Word2Vec or GloVe.

5. Fine-tuning: After pre-training on large text corpora, BERT can be fine-tuned on downstream NLP tasks such as text classification, named entity recognition, question answering, and sentiment analysis. Fine-tuning involves training the model on task-specific datasets with labeled examples, allowing it to adapt to the specific characteristics of the task.

Q11. What is Decoder Only Models?  
Ans: Decoder models use only the decoder of a Transformer model. At each stage, for a given word the attention layers can only access the words positioned before it in the sentence. These models are often called auto-regressive models.

The pretraining of decoder models usually revolves around predicting the next word in the sentence. These models are best suited for tasks involving text generation.

Q12. What is the working of a transformer?  
Ans: The input is a prompt (often referred to as context) fed into the transformer as a whole. There is no recurrence.

The output depends on the goal of the model. For GPT models, the output is a probability distribution of the next token/word that comes after the prompt. It outputs one prediction for the complete input.

Next, it is essential to understand the key components that make up the decoder-only transformer architecture:

The embedding: the input of the transformer model is a prompt. This prompt needs to be embedded into something that the model can use.

The block(s): This is the main source of complexity. Each block contains a masked multi-head attention submodule, a feedforward network, and several layer normalization operations. Blocks are put in sequence to make the model deeper.

The output: the output of the last block is fed through one more linear layer to obtain the final output of the model (a classification, a next word/token etc.)

Q13. Write down the working of a self-attention model.

Ans: Working of a self-attention model is as follows-  
Simplified explanation of how an attention model works-

1. **Input Representation:** Assume you have a sequence of input elements (words in a sentence, pixels in an image, etc.).
2. **Encoder:** The input sequence is passed through an encoder to create a set of representations (often called encodings or embeddings). This captures the information present in the input.
3. **Attention Mechanism:** The attention mechanism is introduced to dynamically weigh the importance of different parts of the input sequence when making predictions.
4. **Key, Query, and Value:** The attention mechanism typically involves three components: key, query, and value. These are derived from the encoder's output.

**Key (K): Represents the information in the input.**

**Query (Q): Represents what the model is currently looking at or attending to.**

**Value (V): Represents the information associated with each element in the input sequence.**

**5. Attention Scores:** The attention scores are computed by measuring the similarity between the query and key vectors. One common method is to use the dot product, but other methods such as scaled dot-product attention or attention with learnable parameters can be employed.

**6. Softmax and Attention Weights:** The attention scores are passed through a softmax function to obtain attention weights. Softmax ensures that the weights sum up to 1, making them a probability distribution.

**7. Weighted Sum of Values:** The attention weights are used to calculate a weighted sum of the values. This emphasizes the parts of the input sequence that are more relevant for the task at hand.

**8. Context Vector:** The weighted sum of values gives a context vector, which is a refined representation of the input sequence based on the attention mechanism.

**9. Decoder:** The context vector is then used by a decoder to generate the final output. In sequence-to-sequence tasks (like language translation), the decoder might be autoregressive and generate one element at a time.

Q14. What are the components of Language Modeling (LM)?

Ans: Components of Language Modeling are-   
Vocabulary: Set of unique words in the training dataset.

n-grams: Sequences of n words used to model language structure.

Probability Distribution: Assigns probabilities to different word sequences

Q15. What are neural language models?  
Ans: Neural language models (NLMs) are a class of language models that use neural networks to represent and generate natural language text. They are a type of probabilistic model that learns the probability distribution of sequences of words in a language. Neural language models have become increasingly popular due to their ability to capture complex patterns and dependencies in text data.

Here are some key characteristics and components of neural language models:

1. Word Embeddings: Neural language models typically represent words as dense, low-dimensional vectors known as word embeddings. These embeddings are learned as part of the training process and capture semantic and syntactic similarities between words. Popular techniques for learning word embeddings include Word2Vec, GloVe, and FastText.

2. Recurrent Neural Networks (RNNs): Early neural language models often used recurrent neural networks (RNNs) as the underlying architecture. RNNs are well-suited for sequential data processing tasks, making them suitable for modeling language. RNN-based language models process input sequences one word at a time, updating their hidden state at each step based on the current word and the previous hidden state.

3. Long Short-Term Memory (LSTM) Networks: To address the vanishing gradient problem and capture long-range dependencies more effectively, many neural language models use variants of RNNs such as Long Short-Term Memory (LSTM) networks or Gated Recurrent Units (GRUs). These architectures allow the model to retain information over longer sequences, enabling better performance on tasks such as language modeling.

4. Transformer Architecture: More recently, transformer-based architectures have gained prominence in the field of neural language modeling. Transformers use self-attention mechanisms to capture contextual information from input sequences efficiently. Models such as BERT, GPT, and T5 are based on the transformer architecture and have achieved state-of-the-art performance on various NLP tasks.

5. Training Objective: Neural language models are typically trained using maximum likelihood estimation (MLE) or its variants, such as cross-entropy loss. The objective is to maximize the probability of generating the next word in a sequence given the previous words. During training, the model adjusts its parameters to minimize the difference between the predicted word distribution and the actual word distribution.

6. Fine-Tuning: Pre-trained neural language models can be fine-tuned on downstream tasks by further training them on task-specific datasets. Fine-tuning allows the model to adapt to the specific characteristics of the task and often leads to improved performance on tasks such as text classification, named entity recognition, and machine translation.  
  
  
MULTIPLE CHOICE QUESTIONS-

Q1. What does NLP stand for?

a) Natural Language Production

b) Neural Language Processing

c) Natural Language Processing

d) Nonlinear Linguistic Patterns

Correct Answer: c) Natural Language Processing

Q2. Which of the following is a task in NLP?

a) Image classification

b) Speech recognition

c) Object detection

d) Video segmentation

Correct Answer: b) Speech recognition

Q3. What is the primary objective of a language model (LM)?

a) Translate text from one language to another

b) Generate new text that resembles a given text corpus

c) Classify text into predefined categories

d) Extract entities from text data

Correct Answer: b) Generate new text that resembles a given text corpus

Q4. Which of the following is an example of an encoder-only model?

a) BERT

b) GPT

c) Transformer

d) BiLSTM

Correct Answer: a) BERT

Q5. What does BERT stand for?

a) Bidirectional Encoder Representations from Transformers

b) Bidirectional Encoder Recurrent Transformer

c) Biased Embedding Representations for Text

d) Bidirectional Embedding and Recurrent Text

Correct Answer: a) Bidirectional Encoder Representations from Transformers

Q6. Which component is used to capture context in a decoder-only model?

a) Self-attention mechanism

b) LSTM layer

c) Convolutional layer

d) Fully connected layer

Correct Answer: a) Self-attention mechanism

Q7. What is the purpose of an encoder-only model in NLP?

a) To generate text from scratch

b) To understand the contextual meaning of a sentence

c) To translate text from one language to another

d) To summarize a given text

Correct Answer: b) To understand the contextual meaning of a sentence

Q8. Which model architecture is used in OpenAI's GPT series?

a) Encoder-only

b) Decoder-only

c) Transformer

d) LSTM

Correct Answer: b) Decoder-only

Q9. What is the primary difference between encoder-only and decoder-only models?

a) Encoder-only models generate output sequences one token at a time.

b) Decoder-only models capture context from surrounding words in a sentence.

c) Encoder-only models incorporate self-attention mechanisms.

d) Decoder-only models typically use autoregressive generation.

Correct Answer: d) Decoder-only models typically use autoregressive generation.

Q10. What is the purpose of the self-attention mechanism in transformer models?

a) To calculate the similarity between words in a sentence.

b) To determine the probability of each word in the input sequence.

c) To encode contextual information from the entire input sequence.

d) To generate output tokens autoregressively.

Correct Answer: c) To encode contextual information from the entire input sequence.