Roll No: | | | | | | | | | | | |

PRESIDENCY UNIVERSITY

GAIN MORE KNOWLEDGE
REACH GREATER HEIGHTS

Presidency University Act, 2013 of the Karnataka Act No. 41 of 2013 | Established under Section 2(f) of UGC Act, 1956
Approved by AICTE, New Delhi | Approved By BCI
Bengaluru

Mid Term Examinations - October 2026

Date: 05/10/2026 **Time:** 11:45 AM - 01:15 PM

Course Code: CAI3411 **Course Name:** Generative AI

Semester: Seventh Semester **Max. Marks:** 50 **Weightage:** 50%

CO - Levels	CO1	CO2	CO3
Marks	52	68	60

PART A: Answer Following Questions. 10 M

Q.N.	Questions	M	CO	BT
1	A prompt is the input (instruction, question, context, examples) given to a generative model to obtain an output. Prompt engineering is the practice of designing and refining prompts so that the model gives accurate, relevant and well-formatted responses.	2	CO1	BT1
2	ask is the specific job to do in this request (e.g., "Summarise the text below"). System instruction is the persistent, higher-level guidance that sets the model's role, tone, rules and safety limits (e.g., "You are a polite banking assistant; never reveal account data") and applies to the whole conversation.	2	CO1	BT1
3	Tokenizer, embedding layer, positional encoding, stacked transformer blocks (multi-head self-attention, feed-forward network, residual connections, layer normalization) and an output (softmax) layer, trained on very large corpora with billions of parameters.	2	CO1	BT1
4	A dense, fixed-length numeric vector that represents a token/word/sentence so that similar meanings lie close together. Example: "king" → [0.21, −0.40, 0.87, ...]; vector(king) − vector(man) + vector(woman) ≈ vector(queen).	2	CO1	BT1
5	A generative model trained on millions of images and captions can, given the prompt "an astronaut riding a horse on Mars", synthesize a brand-new image that never existed in the training set. Likewise, GPT given "Write a poem on rain" produces a new poem.	2	CO2	BT2

PART B : Answer Any 1 Following Questions. 10 M

Q.N.	Questions	M	CO	BT

ANSWER KEY

6	ive generative models for different data modalities (in detail): 1. Text-to-text – GPT-4 / PaLM 2 / LLaMA. Transformer decoder trained on huge text corpora by next-token prediction, then aligned with RLHF. Input: text prompt; output: text. Applications: chatbots, summarisation, translation, code writing. 2. Text-to-image – DALL·E 3 / Stable Diffusion / Midjourney. Diffusion-based models; a text encoder (CLIP/T5) conditions a denoising U-Net that turns random noise into an image matching the prompt. Applications: advertising, concept art, game design, product mock-ups. 3. Image-to-image – CycleGAN / pix2pix / SD img2img. GAN or diffusion models that convert one image domain into another (unpaired for CycleGAN, paired for pix2pix). Applications: photo→painting, day→night, sketch→photo, medical image translation. 4. Image-to-text – BLIP / GPT-4V / Flamingo. Vision encoder + language decoder; produces captions or answers questions on images (VQA). Applications: alt-text for visually impaired, image search, document understanding. 5. Audio-to-text – Whisper (OpenAI). Encoder–decoder transformer trained on 680,000 hours of multilingual audio; converts speech to text, translates and identifies language. Applications: subtitles, meeting transcription, voice assistants.	10	CO3	BT3
7	**Four parameters used with generative models:** 1. Temperature (0–2). Scales the logits before softmax. Low temperature sharpens the distribution → the most probable token is nearly always picked (factual QA, code). High temperature flattens it → varied, creative output (story writing) but risk of errors. Example: T=0.2 "The sky is blue"; T=1.2 "The sky is a shimmering canvas of dusk". 2. **Top-K.** Restricts choice to the K highest-probability tokens, then samples among them. K=1 = greedy decoding; K=40 gives moderate diversity; removes very unlikely, nonsensical tokens. 3. Top-P (nucleus). Chooses the smallest group of tokens whose cumulative probability exceeds P (e.g., 0.9). The candidate set size changes with the model's confidence, so it is more adaptive than top-K. Usually tune temperature or top-P, not both aggressively. 4. Max output tokens / length. Sets the maximum tokens generated; prevents overly long output and controls cost/latency; output is truncated when the limit is hit. Related: stop sequences, frequency/presence penalty to avoid repetition. For image models the analogues are guidance scale, steps, seed and image size.	10	CO2	BT3

PART C: Answer Any 1 Following Questions. 10 M				
Q.N.	**Questions**	**M**	**CO**	**BT**
8	GPT (Generative Pre-trained Transformer) – OpenAI's decoder-only transformer that uses masked (causal) self-attention and generates text token by token (autoregressive). Two stages: unsupervised pre-training on large text, then supervised fine-tuning/RLHF. Variants GPT-1 (2018): 117M parameters, showed pre-train + fine-tune works.GPT-2 (2019): 1.5B, coherent long text, zero-shot ability.GPT-3 (2020): 175B, few-shot in-context learning.GPT-3.5 / ChatGPT: instruction-tuned with RLHF for dialogue.GPT-4 / GPT-4o: multimodal (text+image[+audio]), longer context, better reasoning and safety.Codex: GPT fine-tuned on code (GitHub Copilot). Features: transformer decoder, self-attention, large scale, transfer learning, few/zero-shot learning, context awareness, human-feedback alignment, multilingual, tool/plug-in use. Applications: conversational agents, content writing, summarisation, translation, code generation/debugging, tutoring, data extraction, search and analytics, creative writing.	10	CO2	BT2

ANSWER KEY

9	Applications of generative models: 1. Text/NLP – chatbots, content and report writing, summarisation, translation, email drafting, question answering. 2. Code – code completion, bug fixing, test generation (Copilot, Codex). 3. Images & art – text-to-image design, logo/advert creation, style transfer, photo editing, inpainting, super-resolution. 4. Audio & speech – text-to-speech, voice cloning, music composition, speech-to-text (Whisper) 5. Video & animation – text-to-video, video editing, game asset generation. 6. Healthcare & science – drug/molecule design, synthetic medical images, protein structure prediction, medical report summaries. 7. Education – personalised tutoring, quiz creation. 8. Business – customer support agents, marketing copy, data augmentation/synthetic data, product design, financial report analysis. 9. Cyber-security & simulation – generating synthetic attack data, autonomous-driving scenarios. Generative AI thus boosts productivity, creativity and automation across industries, though it demands caution about bias, misinformation and deepfakes.	10	CO3	BT2

PART D: Answer Any 1 Following Questions. 10 M				
Q.N.	**Questions**	**M**	**CO**	**BT**
10	An RNN processes a sequence $x_1 \ldots x_T$ one element at a time and keeps a hidden state h_t acting as memory. Hidden state: $h_t = \tanh(W_{xh} x_t + W_{hh} h_{t-1} + b_h)$Output: $y_t = \text{softmax}(W_{hy} h_t + b_y)$ The same weights W_xh, W_hh, W_hy are shared at all time steps. x1 x2 x3 x4 | | | ▼▼▼▼ [h1] → [h2] → [h3] → [h4] (h0 = 0 at start) | | | ▼▼▼▼ y1 y2 y3 y4 Example (next-word prediction): for "the cat sat", h_1 encodes "the", $h_2 = f(h_1$, "cat"), $h_3 = f(h_2$, "sat"); y_3 predicts "on". Training – BPTT: the loss is the sum of errors at each step; the network is unrolled and gradients flow backwards through time, adding up for the shared weights, then weights are updated by gradient descent. Advantages: handles variable length, remembers context, few parameters. Limitations: vanishing/exploding gradients, sequential (slow), poor long-term memory → solved by LSTM/GRU and transformers.	10	CO3	BT3
11	Tokenization converts text into tokens and IDs.: 1. Word-level: split on whitespace/punctuation. + Simple, meaningful; − huge vocabulary, OOV words, "run/running" unrelated. 2. Character-level: each character a token. + Small vocabulary, no OOV; − long sequences, poor semantics, slow. 3. Subword-level: balance of both: BPE: iterative merging of most frequent adjacent pairs (GPT, LLaMA).WordPiece: merges pairs that maximise the likelihood of the training data (BERT); uses ## prefix.Unigram/SentencePiece: starts with a large vocab and prunes low-probability tokens; language-independent (T5).Byte-level BPE: operates on bytes, so no unknown characters (GPT-2/3/4). 4. Patch tokenization (images) – splitting image into P×P patches (ViT). Steps in practice: normalise text → pre-tokenize → apply tokenizer → add special tokens ([CLS], [SEP],) → map to IDs → embeddings. Subword tokenization is the standard in modern LLMs because it keeps the vocabulary small while handling rare words.	10	CO3	BT3

PART E: Answer Any 1 Following Questions. 10 M				
Q.N.	**Questions**	**M**	**CO**	**BT**

ANSWER KEY

12	What is GPT? GPT stands for Generative Pre-trained Transformer. GPT is a family of large language models developed using the Transformer decoder architecture. GPT follows two major stages: Pre-trainingFine-tuning / task adaptationThe model is first trained on a very large collection of text to learn language patterns. It can then be adapted for different NLP tasks. Basic idea Given a sequence: "Artificial intelligence is"GPT predicts the next token: "transforming"It then uses the generated token to predict the next one. Therefore, GPT performs autoregressive language modeling. 3. GPT Architecture GPT primarily uses the Transformer decoder. Input Text ↓ Tokenization ↓ Token Embeddings + Positional Information ↓ Transformer Decoder Block ↓ Masked Multi-Head Self-Attention ↓ Feed Forward Network ↓ Add & Norm ↓ Repeated N times ↓ Linear Layer ↓ Softmax ↓ Next Token Prediction BERT BERT (Bidirectional Encoder Representations from Transformers) is a Transformer-based model mainly designed for understanding language. It uses the Transformer encoder architecture. Architecture Input Text ↓ Token + Position + Segment Embeddings ↓ Bidirectional Self-Attention ↓ Feed Forward Network ↓ Transformer Encoder Layers ↓ Task-specific Output Working BERT is trained mainly using Masked Language Modeling (MLM). For example: "The student is [MASK] in the class."BERT predicts the missing word based on the context on both sides. This is why BERT is called bidirectional. The original BERT training also used Next Sentence Prediction (NSP). Applications Sentiment analysisText classificationNamed Entity RecognitionQuestion answeringSentence similarityInformation extraction	10	CO3	BT3

ANSWER KEY

13	RAG Architecture and Components RAG (Retrieval-Augmented Generation) is an NLP technique that combines information retrieval with Large Language Models (LLMs) to generate answers using relevant external knowledge. RAG Architecture User Query ↓ Query Processing ↓ Embedding Model ↓ Vector Database ↓ Relevant Documents ↓ Context + Query ↓ LLM ↓ Generated Answer Main Components Document Collection Contains PDFs, webpages, databases, documents, etc.Provides the knowledge used by the RAG system.Document Chunking Large documents are divided into smaller pieces called chunks.This makes retrieval more efficient.Embedding Model Converts text/chunks into numerical vectors called embeddings.Similar meanings produce similar vector representations.Vector Database Stores document embeddings.Examples: FAISS, Chroma, Pinecone, Weaviate.Used to find documents similar to the user's query.Retriever Converts the user's query into an embedding.Searches the vector database and retrieves the most relevant chunks.Prompt / Context Construction The retrieved information is combined with the user's query.This creates the context given to the LLM.LLM (Large Language Model) Uses the query and retrieved context to generate the final response.Generated Response The LLM produces an answer based on the retrieved information. Simple Example Suppose a university has 100 PDFs containing regulations. User asks: "What is the minimum attendance required for the PhD coursework?"RAG: Question ↓ Embedding ↓ Search University PDFs ↓ Retrieve relevant regulation ↓ Give retrieved text + question to LLM ↓ Generate Answer Key Advantage The main advantage of RAG is that the LLM can use external and domain-specific information without necessarily retraining the entire model. In short: RAG = Retrieve relevant information + Generate an answer using that information.	10	CO3	BT3

ANSWER KEY