Topic 2: Data Preprocessing & Embedding Generation

MCQs (26–50)

A. Basics of Embeddings

Q26. What is the primary purpose of **embeddings** in a RAG pipeline?
a) To store raw text efficiently
b) To represent text numerically in a semantic space
c) To compress documents
d) To fine-tune LLMs
Answer: b

Q27. Which similarity measure is most commonly used for comparing embeddings?
a) Euclidean distance
b) Cosine similarity
c) Jaccard similarity
d) Manhattan distance
Answer: b

Q28. Before generating embeddings, why is **data preprocessing** important?
a) To reduce embedding dimensionality
b) To remove noise, duplicates, and irrelevant text
c) To change the embedding model's architecture
d) To train a language model
Answer: b

Q29. If documents contain HTML tags, what's the best step before embedding generation?
a) Use a PDF loader
b) Use `UnstructuredHTMLLoader` or BeautifulSoup to clean text
c) Skip preprocessing
d) Use OpenAI API directly
Answer: b

Q30. Which of the following can **negatively impact** embedding quality?
a) Duplicate documents
b) Uncleaned HTML
c) Noisy OCR text
d) All of the above
Answer: d

B. OpenAI Embeddings

Q31. Which OpenAI model is best suited for **high-accuracy semantic search**?
a) `text-embedding-ada-002`
b) `text-embedding-3-large`
c) `gpt-4`
d) `davinci`
Answer: b

Q32. Compared to `text-embedding-3-large` , the `text-embedding-3-small` model offers:
a) Higher accuracy and higher cost
b) Lower accuracy and lower cost
c) Lower accuracy but higher cost
d) Identical accuracy but slower speed
Answer: b

Q33. OpenAI embeddings are generally represented as:
a) Sparse matrices
b) Dense vectors
c) Hash maps
d) Graph embeddings
Answer: b

Q34. What is the typical **dimension size** of OpenAI's `text-embedding-3-large` model?
a) 384
b) 512
c) 1536

d) 4096
Answer: c

Q35. OpenAI embeddings are typically integrated into RAG pipelines via:
a) REST APIs
b) LangChain
c) LlamaIndex
d) All of the above
Answer: d

C. Hugging Face Sentence Transformers

Q36. Which Hugging Face model is most commonly used for **lightweight semantic search**?
a) `all-MiniLM-L6-v2`
b) `e5-large`
c) `roberta-large`
d) `gpt2-xl`
Answer: a

Q37. Compared to `all-MiniLM-L6-v2`, the `e5-large` model is:
a) Faster but less accurate
b) Slower but more accurate
c) Similar in both speed and accuracy
d) Smaller in size
Answer: b

Q38. Which library is commonly used to implement Hugging Face embeddings?
a) `sentence-transformers`
b) `torchvision`
c) `transformers-only`
d) `openai`
Answer: a

Q39. If multilingual document retrieval is required, which Hugging Face embedding model is best?
a) `all-MiniLM-L6-v2`

b) `distilbert-base-uncased`
c) `LaBSE` or multilingual `e5` models
d) `gpt-j`
Answer: c

Q40. Hugging Face embeddings are generated using:
a) Token-level attention
b) CLS token pooled representations
c) Sentence-level transformers
d) All of the above
Answer: d

D. Cohere & Instructor Embeddings

Q41. Cohere embeddings are particularly known for:
a) Multilingual capabilities
b) Image captioning
c) OCR-based embedding
d) Audio processing
Answer: a

Q42. Which Cohere API endpoint is used to generate embeddings?
a) `/v1/embeddings`
b) `/generate`
c) `/classify`
d) `/multilingual`
Answer: a

Q43. Instructor embeddings differ from standard embeddings because they:
a) Require more tokens
b) Are **task-specific** and take instructions to generate embeddings
c) Are less accurate
d) Work only in English
Answer: b

Q44. An example model for Instructor embeddings is:
a) `hkunlp/instructor-large`

b) `text-embedding-3-large`
c) `cohere-multilingual-v3`
d) `sentence-transformers/all-MiniLM-L6-v2`
Answer: a

Q45. When working on domain-specific retrieval, Instructor embeddings are preferred because:
a) They require fewer resources
b) They allow embedding customization per task
c) They bypass vector databases
d) They do not require chunking
Answer: b

E. Integrating Embeddings into Pipelines

Q46. LangChain integrates embeddings by using which class?
a) `EmbeddingsPipeline`
b) `OpenAIEmbeddings` or `HuggingFaceEmbeddings`
c) `EmbeddingsManager`
d) `VectorLoader`
Answer: b

Q47. LlamaIndex embeddings can be combined with which indexes for retrieval?
a) `VectorStoreIndex`
b) `TreeIndex`
c) `ListIndex`
d) All of the above
Answer: d

Q48. If a project involves multiple embedding models for different languages, which LangChain feature can help?
a) Multi-embedding routing
b) Model chaining
c) Recursive embedding pipelines
d) Parallel embeddings
Answer: a

Q49. Which of the following is TRUE about embeddings integration in RAG?
a) Embeddings are always stored in relational databases
b) Embeddings are stored in **vector databases** for similarity search
c) Embeddings replace retrievers
d) Embeddings cannot be cached
Answer: b

Q50. Which pipeline correctly represents embedding integration?
a) Load → Embed → Store → Retrieve → Generate
b) Load → Store → Generate → Embed
c) Embed → Load → Generate → Store
d) Generate → Embed → Load → Retrieve
Answer: a