Topic 4: Handling Longtail Queries

MCQs (76–100)

A. Basics of Longtail Queries

Q76. In RAG, what are **longtail queries**?
a) Queries containing very common keywords
b) Queries with highly specific or niche information needs
c) Queries limited to single keywords only
d) Queries unrelated to embeddings
Answer: b

Q77. Why do **regular vector searches** often fail for longtail queries?
a) Embeddings ignore rare entities or domain-specific terms
b) Vector DBs cannot handle high dimensions
c) Index size limitations
d) Embedding dimensionality mismatch
Answer: a

Q78. Which retrieval strategy improves longtail query performance by **combining semantic and lexical search**?
a) Pure keyword search
b) Hybrid search
c) Sparse retrieval
d) Embedding compression
Answer: b

Q79. If a query includes **rare startup-specific jargon**, what is the best approach?
a) Use low-dimensional embeddings
b) Use Instructor embeddings for **task-specific context**
c) Use OpenAI GPT-3 without retrieval
d) Skip embeddings entirely
Answer: b

Q80. Which of the following is a real-life example of a **longtail query**?
a) "Best Python jobs in India"
b) "NLP engineer roles in Bangalore startups using **Swarm Learning**"
c) "Top data science salaries in the US"
d) "AI jobs"
Answer: b

B. Improving Retrieval for Longtail Queries

Q81. Which of these is **NOT** a good solution for handling longtail queries?
a) Hybrid retrieval
b) Using cross-encoder rerankers
c) Domain-specific embeddings
d) Ignoring chunking and using full documents
Answer: d

Q82. Cross-encoder rerankers improve retrieval by:
a) Re-indexing documents
b) Re-evaluating top-k results from vector search using deeper semantic understanding
c) Compressing embeddings
d) Skipping similarity scores
Answer: b

Q83. When dealing with **niche domains** like genomics or swarm learning, which embeddings are best suited?
a) OpenAI generic embeddings
b) Task-specific embeddings like **Instructor embeddings**
c) Default BM25 sparse embeddings
d) TF-IDF sparse encoders
Answer: b

Q84. Instructor embeddings improve longtail retrieval by:
a) Using predefined instructions to tailor embeddings for specific tasks
b) Reducing embedding dimensionality
c) Skipping vector similarity search

d) Switching to relational databases
Answer: a

Q85. To improve retrieval for **longtail multilingual queries**, which embeddings perform best?
a) `text-embedding-3-small`
b) HuggingFace multilingual `e5-large`
c) TF-IDF vectors
d) Keyword-only search
Answer: b

C. Hybrid Search & Advanced Reranking

Q86. Why does **hybrid search** outperform pure vector search for longtail queries?
a) It reduces embedding size
b) Combines **semantic similarity** with **keyword matching** for better coverage
c) Ignores rare tokens entirely
d) Uses fewer retrieval passes
Answer: b

Q87. Which vector database provides **built-in hybrid search** optimized for longtail queries?
a) FAISS
b) Pinecone
c) Weaviate
d) ChromaDB
Answer: c

Q88. In hybrid retrieval, BM25 contributes by:
a) Ranking results based on exact keyword matches
b) Generating embeddings
c) Chunking documents
d) Removing stopwords automatically
Answer: a

Q89. A typical RAG pipeline for **longtail queries** might follow this order:
a) Ingest → Embed → Hybrid Search → Cross-Encoder Rerank → Generate

b) Embed → Generate → Chunk → Hybrid Search
c) Retrieve → Embed → Generate → Hybrid Search
d) Chunk → Generate → Store → Embed
Answer: a

Q90. Which reranking model is commonly used for **longtail retrieval scenarios**?
a) `ms-marco-MiniLM-L-6-v2`
b) `text-embedding-ada-002`
c) `all-mpnet-base-v2`
d) `gpt2`
Answer: a

D. Human-in-the-Loop Validation & Argilla

Q91. Why is **human-in-the-loop validation** important for longtail queries?
a) Automates all RAG steps
b) Ensures retrieved documents are contextually correct and relevant
c) Replaces embeddings completely
d) Eliminates reranking
Answer: b

Q92. Argilla is primarily used in RAG pipelines for:
a) Generating embeddings
b) Monitoring, validating, and improving retrieval quality
c) Chunking and indexing
d) Replacing vector DBs
Answer: b

Q93. When using Argilla for **longtail queries**, annotators can:
a) Approve, reject, or edit retrieved passages
b) Train cross-encoders interactively
c) Flag irrelevant retrievals for retraining
d) All of the above
Answer: d

Q94. One key advantage of **Argilla-driven RAG pipelines** is:
a) Fully unsupervised document retrieval

b) Continuous feedback loops to improve embeddings and ranking
c) Faster chunking
d) Smaller embedding sizes
Answer: b

Q95. If embeddings consistently fail for longtail queries, Argilla can:
a) Suggest switching to TF-IDF
b) Collect mislabeled examples to retrain embeddings or retrievers
c) Convert embeddings into keywords
d) Disable hybrid search
Answer: b

E. Edge Cases & Evaluation

Q96. Which metric best evaluates RAG performance for **longtail queries**?
a) BLEU
b) MRR (Mean Reciprocal Rank)
c) Token perplexity
d) Word2Vec accuracy
Answer: b

Q97. If a query asks:
"*List NLP jobs in Bangalore startups using Swarm Learning*"
... which step is **most critical**?
a) Using specialized embeddings trained on startup & NLP jargon
b) Using cosine similarity only
c) Relying on random sampling
d) Ignoring rerankers
Answer: a

Q98. To handle longtail queries in **medical domains**, you should:
a) Use BM25 exclusively
b) Use embeddings fine-tuned on PubMed or biomedical corpora
c) Use a lightweight multilingual model
d) Prefer keyword-only retrieval
Answer: b

Q99. Which combined approach gives the **best results** for longtail queries?
a) Vector search only
b) Keyword search only
c) Hybrid retrieval + task-specific embeddings + cross-encoder reranking
d) Traditional SQL-based queries
Answer: c

Q100. In RAG pipelines, which is the **most effective high-level strategy** for longtail queries?
a) Replace embeddings with BM25
b) Use **hybrid retrieval + advanced rerankers + human feedback**
c) Reduce chunk size drastically
d) Skip embeddings and rely on GPT only
Answer: b