SVO-triplet constrained, unsupervised QA synthesis pipeline with hybrid dual-scale retrieval for technical knowledge indexing

ML/AI demo by Tamas Horvath (txttw)
Last updated: 2025-07-27
git: https://github.com/txttw/qa-synth-pipeline

Overview

This system transforms unstructured technical documentation (product specs, expert articles, manuals) into a query-optimized knowledge base. By integrating **linguistic structure extraction, constrained generation, and multi-strategy retrieval,** it enables precise technical question answering while minimizing computational costs.

Target application

Ideal for knowledge bases, chat-bots, and product search.

Key Components

Semantic-aware chunking

- After coarse rule based chunking with overlap an LLM splits the larger texts into smaller coherent chunks, preserving technical context and logical flow.
- Benefit: Maintains explanatory relationships between concepts within chunks, avoiding arbitrary fragmentation.

SVO triplet extraction

- Each sentence in a chunk is processed to extract Subject-Verb-Object (SVO) triplets (e.g., "The algorithm [S] optimizes [V] performance [O]").
- *Design decision*: Triplets enforce structured representation of core relationships, reducing ambiguity and halucination.

SVO-constrained QA generation

- An LLM generates QA pairs using SVO triplets as constraints, with the full chunk as context.
- *Key Innovations*:
  - Cost efficiency: SVO direction allows smaller/cheaper LLMs (e.g., 7B-parameter models) without sacrificing precision.
  - Detail preservation: Questions target granular technical elements (e.g., "What does the algorithm optimize?") while retaining broader context.

Validation & Filtering

- A “roberta” based model distilled for Extractive QA scores each generated pair:
  - Answer relevance to the question.
  - Contextual alignment with the source chunk.
- Low-scoring QAs are discarded via a threshold.
- *Benefit*: Ensures factual accuracy and eliminates hallucinated content.

Redundancy reduction

- Question embeddings are clustered using Affinity Propagation to identify semantic redundancy.
- *Design decision*: Affinity Propagation adapts to unknown cluster counts, automatically retaining representative questions.

Dual-scale Embedding

- Final QA pairs are embedded jointly (question + answer as one vector). for precise, detail-oriented queries.
- Whole chunks for broad-context retrieval.

Hybrid embedding:

- Dense retrieval: Semantic matching via dense embedding
- Sparse retrieval: Term-attention based matching with IDF calculation for keyword relevant queries
- Late Interaction (ColBERT): Contextualized token-level relevance scoring

Benefits

Cost-effective scalability

- Indexing: Smaller LLMs handle SVO-triplet-constrained generation, reducing indexing costs vs. large-model approaches.
- Retrieval: Only uses embedding, vector-search and a re-ranking. No LLM or NER model.

Precision-recall balance

- SVO anchoring captures atomic details, while chunk context supports conceptual queries.

High signal-to-noise ratio

- Validation + clustering typically removes 30-40% of generated QAs, retaining only novel, high-value pairs.

Optimized retrieval

- Joint QA embeddings enable direct matching of questions to answer-bearing content.
- Dual embeddings support both specific fact lookup and exploratory searches.
- Hybrid (dense+sparse) embedding increase retrieval precision