Embedding Model Evaluation Report

This report provides a comprehensive evaluation of embedding models. The evaluation includes embedding model selection criteria, ground truth generation, chunking strategies, model performance metrics, and a final conclusion identifying the best embedding configuration.

1. Embedding Model Evaluation Criteria

Embedding models were evaluated based on multiple retrieval quality metrics to assess their suitability for semantic search and retrieval-augmented generation (RAG) tasks. The evaluation criteria are as follows:

Metric	Description	Why It Matters
Precision@3	Fraction of retrieved chunks that are relevant in top-3	Measures accuracy of the top results — crucial for RAG.
Recall@3	Fraction of all relevant chunks found within top-3	Shows how much relevant information is captured early.
Precision@5	Fraction of retrieved chunks that are relevant in top-5	Evaluates a slightly broader retrieval scope.
Recall@5	Fraction of all relevant chunks found within top-5	Assesses comprehensiveness of retrieval.
MRR	Mean Reciprocal Rank of first relevant chunk	Reflects ranking quality — the higher, the better.
nDCG@5	Normalized Discounted Cumulative Gain	Measures ranking quality, emphasizing relevant chunks higher in order.

2. Ground Truth Generation

Ground truth chunks were generated automatically using a Large Language Model (LLM) with a human-in-the-loop review process. This approach eliminates the need for manual dataset creation for each evaluation cycle, ensuring scalability and consistency. The LLM identifies the most relevant chunks corresponding to specific queries, producing reliable ground truth for model evaluation.

Ground truth data is formatted using TOON (Token Object Oriented Notation), a lightweight and token-efficient alternative to JSON. TOON minimizes syntactic overhead and reduces API costs while maintaining human readability, achieving 30–60% token savings.

3. Chunking Strategies

Two chunking strategies were employed to prepare the document text for embedding and retrieval evaluation:

1. Structured Chunking — Based on document headings hierarchy, preserving semantic structure and contextual integrity.

2. Recursive Character Chunking — Splits text recursively by character count, suitable for large text blocks or unstructured data.

Structured chunking generally yields higher retrieval accuracy because it maintains section-level context and meaning, whereas recursive chunking can sometimes fragment related ideas across multiple chunks.

4. Shortlisted Embedding Models

Model	Description
OpenAI - text-embedding-3-small	High-quality, general-purpose embeddings for semantic similarity and retrieval tasks.
Cohere - embed-v4.0	Optimized for robust semantic retrieval and information retrieval use cases.
Hugging Face - all-MiniLM-L6-v2	Lightweight and efficient embeddings for smaller-scale applications.

5. Embedding Model Evaluation Results

Recursive Chunking Evaluation Results:

Model	Recall@3	Recall@5	Precision@3	Precision@5	MRR	nDCG@3	nDCG@5
OpenAI	0.6333	0.80	0.5333	0.40	0.75	0.6325	0.7021
Cohere	0.70	0.80	0.60	0.40	0.8667	0.7370	0.7641
Open Source	0.45	0.70	0.40	0.36	0.75	0.4938	0.6173

Best Model (Recursive Chunking): Cohere (embed-v4.0) with highest MRR (0.8667) and nDCG@5 (0.7641).

Structured Chunking Evaluation Results:

Model	Recall@3	Recall@5	Precision@3	Precision@5	MRR	nDCG@3	nDCG@5
OpenAI	0.7833	0.90	0.60	0.44	1.00	0.8757	0.9161
Cohere	0.6833	0.75	0.5333	0.36	1.00	0.7983	0.8051
Open Source	0.6833	0.75	0.5333	0.36	0.8667	0.7370	0.7438

Best Model (Structured Chunking): OpenAI (text-embedding-3-small) with perfect MRR (1.00) and highest nDCG@5 (0.9161).

6. Conclusion

The evaluation demonstrates that **Structured Chunking combined with OpenAI's text-embedding-3-small** model delivers the **best overall retrieval accuracy and ranking quality**.

Why It Performs Best:

- Structured chunking preserves section-level context.
- Excellent top-ranked relevance reflected by perfect MRR.

Recommended Enhancements:

1. **Metadata Filtering:**
   Narrow searches using section names or tags to improve precision.
2. **Knowledge Graphs:**
   Link concepts and entities for relationship-based retrieval.
3. **Hybrid Search:**
   Combine vector and keyword searches for balanced precision and recall.
4. **Query Expansion:**
   Use synonym or concept expansion to improve recall coverage.

Code Repository: https://github.com/JS12540/evaluating_embedding_models