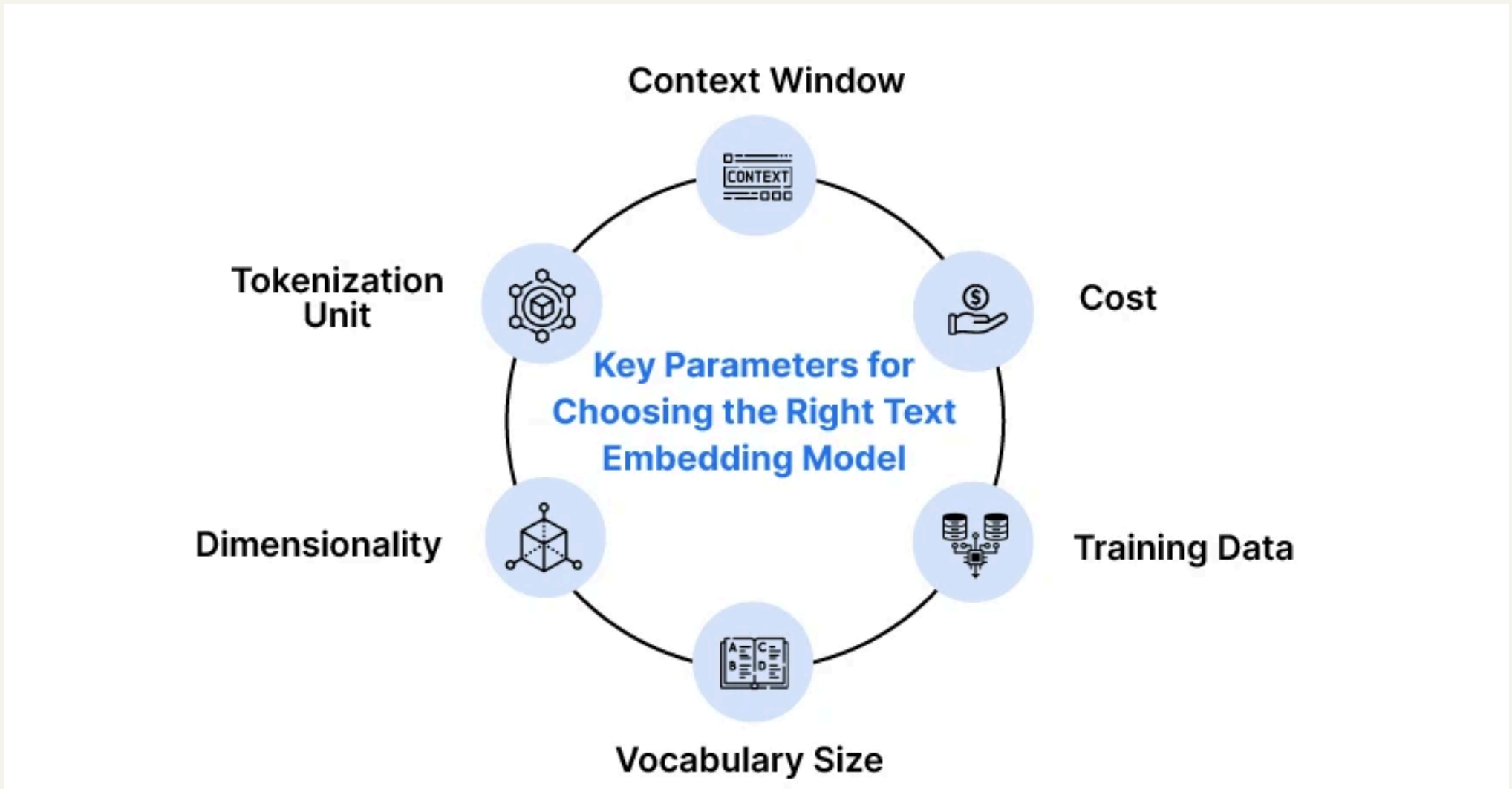
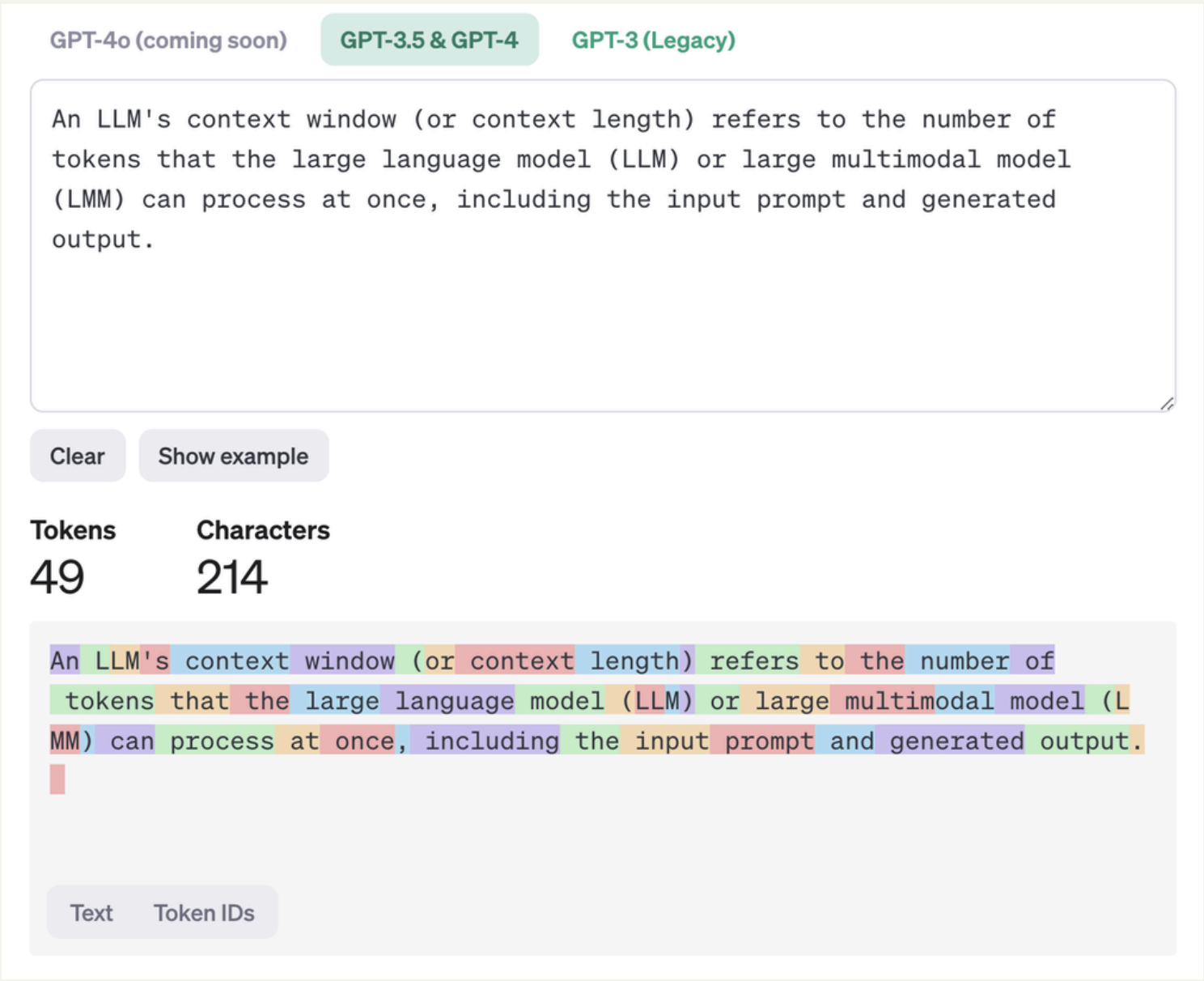
How to choose the best Embedding Model for RAG Systems

A Practical Guide



Dipanjan (DJ)

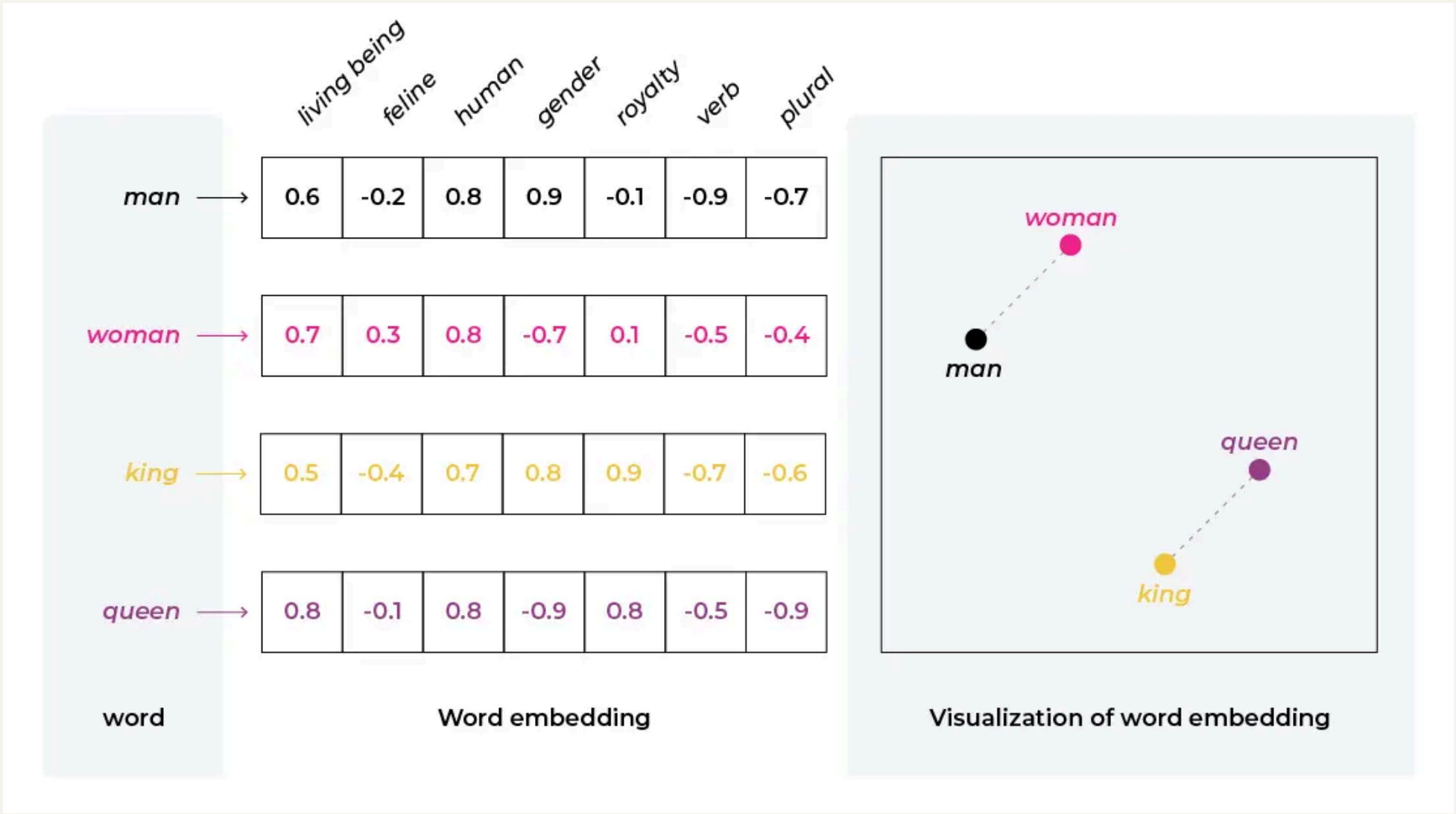
Key Factors for Choosing the best Embedding Model



- **Context Window**
  - A context window is the largest number of tokens (words or subwords) a model can process in one input. For instance, If a model has a context window of 512 tokens, it can only process 512 tokens at a time. Longer texts will be truncated or would need to be split into smaller chunks
  - A wider context window enables the model to handle longer documents or strings of text without being cut off.
  - For operations such as semantic search on long texts (e.g., scientific articles), a big context window (e.g., 8192 tokens) is required.
- **Tokenization Unit**
  - Tokenization is breaking down text into smaller units (tokens) that the model can process. The tokenization unit refers to the method used to split text into tokens
  - The tokenization technique impacts the quality of how the model processes text, particularly for infrequent or specialized words.
  - Most contemporary models favor subword tokenization since it meets both vocabulary size and flexibility needs.

Source: How to Choose the Right Embedding for Your RAG Model

Key Factors for Choosing the best Embedding Model

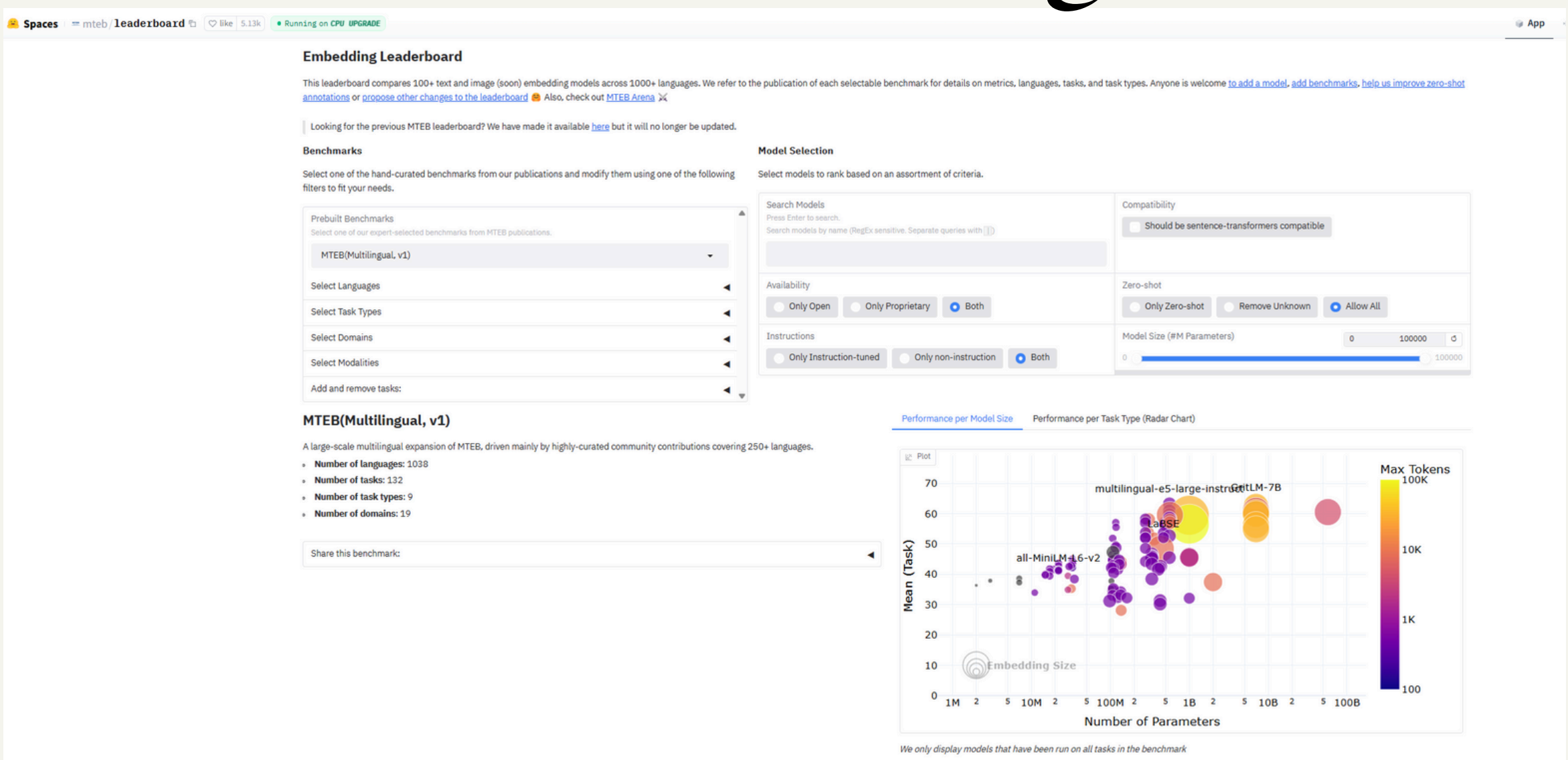


Source: Arize AI

- **Dimensionality**
  - Dimensionality refers to the size of the embedding vector produced by the model. For example, a model with 768-dimensional embeddings outputs a vector of 768 numbers for each input text.
  - Higher-dimensional embeddings can capture more nuanced semantic information but require more computational resources.
  - Lower-dimensional embeddings are faster and more efficient but may lose some semantic richness.
- **Vocabulary Size**
  - The vocabulary size is the number of unique tokens (words or subwords) that the tokenizer can recognize.
  - A larger vocabulary size allows the model to handle a wider range of words and languages but increases memory usage.
  - A smaller vocabulary size is more efficient but may struggle with rare or domain-specific terms.

Source: How to Choose the Right Embedding for Your RAG Model

Key Factors for Choosing the best Embedding Model



Source: Hugging Face MTEB

- **Training Data**
  - Training data refers to the dataset used to train the model. It determines the model's knowledge and capabilities.
  - The quality and diversity of the training data affect the model's performance.
- **Cost**
  - Cost refers to the financial and computational resources required to use an embedding model, including expenses related to infrastructure, API usage, and hardware acceleration.
  - API-based models are convenient but can become expensive for large-scale applications and Open-source models are cost-effective but require technical expertise and infrastructure setup.
- **Quality (Benchmark Performance)**
  - The MTEB (Massive Text Embedding Benchmark) score measures the performance of an embedding model across a wide range of tasks, including semantic search, classification, and clustering.
  - Test on your own dataset and queries and evaluate performance

Source: How to Choose the Right Embedding for Your RAG Model

Popular Embedding Models

Model	Context Window	Cost (per 1M tokens)	Quality (MTEB Score)	Vocab Size	Tokenization Unit	Dimensionality	Training Data
OpenAI text-embedding-ada-002	8192 tokens	$0.10	~61.0	Not publicly disclosed	Subword (Byte Pair)	1536	OpenAI has not publicly disclosed the specific datasets used to train this model.
NVIDIA NV-Embed-v2	32768 tokens	Open-source	72.31	50,000+	Subword (Byte Pair)	4096	Trained using hard-negative mining, synthetic data generation, and existing publicly available datasets.
OpenAI text-embedding-3-large	8192 tokens	$0.13	~64.6	Not publicly disclosed	Subword (Byte Pair)	3072	OpenAI has not publicly disclosed the specific datasets used to train this model.
OpenAI text-embedding-3-small	8192 tokens	$0.02	~62.3	50,257	Subword (Byte Pair)	1536	OpenAI has not publicly disclosed the specific datasets used to train this model.
Gemini text-embedding-004	2048 tokens	Not available	~60.8	50,000+	Subword (Byte Pair)	768	Training data not publicly disclosed.
Jina Embeddings v3	8192 tokens	Open-source	~59.5	50,000+	Subword (Byte Pair)	1024	Trained on large-scale web data, books, and other text corpora.
Cohere embed-english-v3.0	512 tokens	$0.10	~64.5	50,000+	Subword (Byte Pair)	1024	Trained on large-scale web data, books, and other text corpora.
voyage-3-large	32000 tokens	$0.06	~60.5	50,000+	Subword (Byte Pair)	2048	Trained on diverse datasets across multiple domains, including large-scale web data, books, and other text corpora.
voyage-3-lite	32000 tokens	$0.02	~59.0	50,000+	Subword (Byte Pair)	512	Trained on diverse datasets across multiple domains, including large-scale web data, books, and other text corpora.
Stella 400M v5	512 tokens	Open-source	~58.5	50,000+	Subword (Byte Pair)	1024	Trained on large-scale web data, books, and other text corpora.
Stella 1.5B v5	512 tokens	Open-source	~59.8	50,000+	Subword (Byte Pair)	1024	Trained on large-scale web data, books, and other text corpora.
ModernBERT Embed Base	512 tokens	Open-source	~57.5	30,000	WordPiece	768	Trained on large-scale web data, books, and other text corpora.
ModernBERT Embed Large	512 tokens	Open-source	~58.2	30,000	WordPiece	1024	Trained on large-scale web data, books, and other text corpora.
BAAI/bge-base-en-v1.5	512 tokens	Open-source	~60.0	30,000	WordPiece	768	Trained on large-scale web data, books, and other text corpora.

Rank (Borda)	Model	Zero-shot	Number of Parameters	Embedding Dimensions	Max Tokens	Mean (Task)	Mean (TaskType)	Bitext Mining
1	gemini-embedding-exp-03-07	99%	Unknown	3072	8192	**68.32**	**59.64**	79.28
2	Linq-Embed-Mistral	99%	7B	4096	32768	61.47	54.21	70.34
3	gte-Qwen2-7B-instruct	⚠ NA	7B	3584	32768	62.51	56.00	73.92
4	multilingual-e5-large-instruct	99%	560M	1024	514	63.23	55.17	**80.13**
5	SFR-Embedding-Mistral	96%	7B	4096	32768	60.93	54.00	70.00
6	GritLM-7B	99%	7B	4096	4096	60.93	53.83	70.53
7	text-multilingual-embedding-002	99%	Unknown	768	2048	62.13	54.32	70.73
8	GritLM-8x7B	99%	57B	4096	4096	60.50	53.39	68.17
9	e5-mistral-7b-instruct	99%	7B	4096	32768	60.28	53.18	70.58
10	Cohere-embed-multilingual-v3.0	⚠ NA	Unknown	512	Unknown	61.10	53.31	70.50

Source: How to Choose the Right Embedding for Your RAG Model

Read the Practical Guide

How to Choose the Right Embedding for Your RAG Model

Soumil Jain
Last Updated : 18 Mar, 2025

10 min read

Imagine a journalist piecing together a story—not just relying on memory but searching archives and verifying facts. That's how a Retrieval-Augmented Generation (RAG) model works, retrieving real-time knowledge for better accuracy. Just like strong research skills, choosing the best embedding for the RAG model is also crucial for retrieving and ranking relevant information. The right embedding ensures precise and relevant retrieval, enhancing the model's output. The selection of the optimal embedding for RAG models depends on domain specificity, retrieval accuracy, and model architecture. In this blog, we'll explore the steps involved in choosing embeddings for RAG models based on specific applications.

Key Parameters for Choosing the Right Text Embedding Model

RAG models are dependent on good-quality text embeddings for efficiently retrieving relevant information. Text embeddings transform text into numerical values, enabling the model to process and compare text data. The selection of an appropriate embedding model is critical in enhancing retrieval accuracy, response relevance, and overall system performance.

Before jumping into the mainstream embedding models, let's begin by understanding the most important parameters that determine their efficiency. In language model comparison, the key factors to be considered are context window, cost, quality (measured in terms of MTEB score), vocabulary size, tokenization scheme, dimensionality, and training data type. All these factors decide the efficiency, accuracy, and adaptability of a model to different tasks.

New Feature Beta
Personalized GenAI Learning Path 2025✨
Crafted Just for YOU!
Download Now

CHECK OUT THE PRACTICAL GUIDE HERE