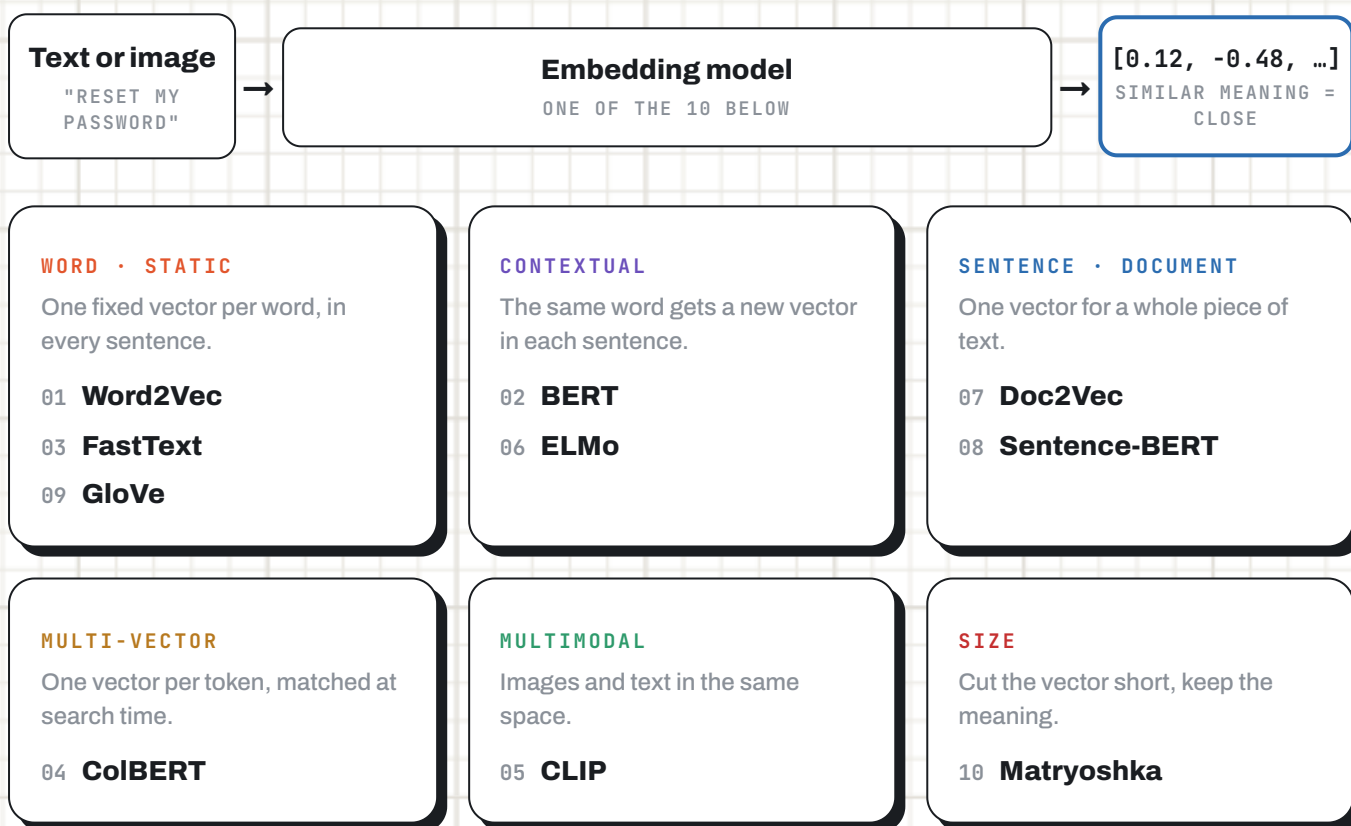


10 Embedding Techniques

HOW TEXT BECOMES A VECTOR

From Word2Vec to Matryoshka: what each one turns into a vector, what it fixed, and which one to use in your RAG system.



The cheat sheet.

ONE LINE EACH

#	TECHNIQUE	FAMILY	A VECTOR PER	IN ONE LINE
01	Word2Vec	WORD	WORD	Learns word vectors by predicting a word's neighbours. Similar contexts, close vectors.
02	BERT	CONTEXTUAL	TOKEN	A transformer reads the whole sentence, so each word's vector depends on its context.
03	FastText	WORD	WORD	Builds a word from its character n-grams, so unseen words and typos still get a vector.
04	CoBERT	MULTI-VECTOR	TOKEN	Keeps a vector for every token and matches query and passage token by token.
05	CLIP	MULTIMODAL	IMAGE / TEXT	Trains an image encoder and a text encoder so a picture and its caption land together.
06	ELMo	CONTEXTUAL	TOKEN	Context-aware word vectors from two LSTM language models, one reading each way.
07	Doc2Vec	DOCUMENT	DOCUMENT	Word2Vec plus a learned vector for each document.
08	Sentence-BERT	SENTENCE	SENTENCE	BERT fine-tuned on sentence pairs: one vector per sentence, compared with cosine.
09	GloVe	WORD	WORD	Fits word vectors to how often words appear together across the whole corpus.
10	Matryoshka	SIZE	TEXT	Trains one vector so its first 64, 128 or 256 numbers each work on their own.

The one idea to keep: an embedding turns text (or an image) into a list of numbers where similar meanings sit close together. The field moved from one vector per word (**01, 03, 09**), to vectors that change with context (**06, 02**), to one vector per sentence built for search (**07, 08**). Most RAG retrievers today use a sentence encoder.

How to use this guide: read the cheat sheet, then the card for anything you couldn't explain out loud. Each card ends with a one-line interview answer and a link to the original paper.

01 Word2Vec

WORD · STATIC

WHAT IT IS

A shallow neural network trained to predict a word from its neighbours (CBOW) or the neighbours from a word (skip-gram). The weights it learns become the word vectors.

WHY IT MATTERS

Showed that meaning can be arithmetic: king – man + woman ≈ queen. The limit: one vector per word, so "bank" by a river and "bank" with a loan share a vector, and unseen words get none.

Typical: 100–300 dimensions · window ≈ 5 words

SAY IT "Predict a word from its neighbours; the weights you learn are the vectors."

Mikolov et al., 2013 · Efficient Estimation of Word Representations in Vector Space · arxiv.org/abs/1301.3781

02 BERT Bidirectional Encoder Representations from Transformers

CONTEXTUAL

WHAT IT IS

A transformer encoder pre-trained by hiding 15% of the tokens and predicting them from both sides. Every token's output vector depends on the whole sentence.

WHY IT MATTERS

"Bank" now gets a different vector in each sentence. It's the base of most modern text encoders. But raw BERT vectors compare sentences poorly; that's why Sentence-BERT (08) exists.

BERT-base: 12 layers · 768 dims · 110M params

SAY IT "A transformer reads the whole sentence both ways, so each word's vector depends on its context."

Devlin et al., 2018 · BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding · arxiv.org/abs/1810.04805

03 FastText

WORD · STATIC

WHAT IT IS

Word2Vec, but a word's vector is the sum of vectors for its character n-grams (3 to 6 characters), plus the whole word.

WHY IT MATTERS

Words never seen in training, rare words and typos still get a sensible vector from their pieces. Strong for languages with many word forms, like Hindi, Finnish or Turkish.

"where", n = 3 → <wh · whe · her · ere · re>

SAY IT "A word is the sum of its pieces, so unseen words still get a vector."

Bojanowski et al., 2017 · Enriching Word Vectors with Subword Information · arxiv.org/abs/1607.04606

04 ColBERT Contextualized Late Interaction over BERT

MULTI-VECTOR

WHAT IT IS

Encodes the query and each passage separately, but keeps one vector per token. At search time each query token finds its best-matching passage token (MaxSim), and the scores are summed.

WHY IT MATTERS

Close to reranker accuracy, yet passages are still encoded ahead of time. The cost is storage, a vector for every token; ColBERTv2 cut that by 6–10×.

score = Σ over query tokens of max similarity

SAY IT "Keep a vector per token, and match token by token at query time: late interaction."

Khattab & Zaharia, 2020 · ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT · arxiv.org/abs/2004.12832

05 CLIP Contrastive Language–Image Pre-training

MULTIMODAL

WHAT IT IS

An image encoder and a text encoder trained together on image–caption pairs. A contrastive loss pulls each image toward its own caption and away from all the others in the batch.

WHY IT MATTERS

Images and text share one vector space: search photos with words, classify an image by comparing it with "a photo of a {label}", or retrieve images in multimodal RAG.

400M image–text pairs · contrastive loss

SAY IT "Two encoders trained so a picture and its caption land at the same spot."

Radford et al., 2021 · Learning Transferable Visual Models From Natural Language Supervision · arxiv.org/abs/2103.00020

06 ELMo Embeddings from Language Models

CONTEXTUAL

WHAT IT IS

Two LSTM language models, one reading left to right and one right to left. A word's vector is a learned mix of their layers, computed from the whole sentence.

WHY IT MATTERS

The first widely used contextual word vectors: added to existing models, it improved the state of the art on six NLP tasks. BERT replaced it later the same year.

Forward + backward LSTMs · mix of all layers

SAY IT "Read the sentence both ways with LSTMs and mix the layers: a vector per word, in context."

Peters et al., 2018 · Deep contextualized word representations · arxiv.org/abs/1802.05365

07 Doc2Vec Paragraph Vector

DOCUMENT

WHAT IT IS

Word2Vec plus one extra vector per document, trained alongside the word vectors to help predict the document's words. That vector learns what the document is about.

WHY IT MATTERS

The first simple way to embed whole documents, still cheap for clustering a fixed collection. But a new document needs extra training steps to get its vector, and transformer encoders usually beat it.

PV-DM: doc vector + context → next word

SAY IT "Train a vector per document next to the word vectors; it learns the topic."

Le & Mikolov, 2014 · Distributed Representations of Sentences and Documents · arxiv.org/abs/1405.4053

08 Sentence-BERT SBERT

SENTENCE

WHAT IT IS

BERT fine-tuned in a siamese (twin) setup on sentence pairs, with its token vectors averaged into one vector per sentence. Two sentences are compared with cosine similarity.

WHY IT MATTERS

The bi-encoder pattern behind almost every RAG retriever: embed documents once, embed the query, compare. Comes with the sentence-transformers library.

10,000 sentences: 65 hours → ~5 seconds

SAY IT "Fine-tune BERT on pairs so one pooled vector per sentence can be compared with cosine."

Reimers & Gurevych, 2019 · Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks · arxiv.org/abs/1908.10084

09 GloVe Global Vectors

WORD · STATIC

WHAT IT IS

Counts how often every pair of words appears together across the whole corpus, then learns vectors whose dot product matches the log of that count.

WHY IT MATTERS

Uses global counts, not just local windows. Its pre-trained vectors (Wikipedia, Common Crawl) became a standard starting point. Static, like Word2Vec: one vector per word.

$$\mathbf{w}_i \cdot \tilde{\mathbf{w}}_j + \mathbf{b}_i + \tilde{\mathbf{b}}_j \approx \log X_{ij}$$

SAY IT "Count co-occurrences across the whole corpus, then fit vectors to the log counts."

Pennington, Socher & Manning, 2014 · GloVe: Global Vectors for Word Representation · aclanthology.org/D14-1162

10 Matryoshka Matryoshka Representation Learning

SIZE

WHAT IT IS

Trains one embedding so its first 64, 128, 256... dimensions each work on their own, by adding up the loss at every one of those lengths. Like nesting dolls, the small vector sits inside the big one.

WHY IT MATTERS

Cut vectors short to save storage and speed up search, then re-rank with the full length. OpenAI's text-embedding-3 "dimensions" setting works this way.

1,024 dims → first 256: 4× smaller

SAY IT "Pack the most meaning into the first dimensions, then cut the vector to fit your budget."

Kusupati et al., 2022 · Matryoshka Representation Learning · arxiv.org/abs/2205.13147

Measure it: embed 50 real questions from your users, mark the chunk that answers each one, and check how often it lands in the top 5 (recall@5). Compare models, or vector lengths, on that number, not on a leaderboard.

Which one should you use?

START FROM YOUR TASK

Building **RAG** or semantic search over text?

A sentence encoder 08

Retrieval still **misses the right chunk**, and you can afford storage?

CoBERT 04

Searching **images with text**, or the other way round?

CLIP 05

The vector database is **too big or too slow**?

Matryoshka, cut short 10

Fine-tuning for **classification** or **entity tagging**?

BERT 02

Typos, **rare words**, or no GPU at all?

FastText 03

Word features for a classic ML model, on a tight budget?

GloVe or **Word2Vec** 09 01

ELMo (06) and Doc2Vec (07) are mostly history now. Learn them to explain how we got from word vectors to sentence encoders in an interview.

Try it: semantic search with Sentence-BERT

Embed the documents once, embed the question, and pick the closest. With unit-length vectors, cosine similarity is just a dot product.

$$\cos(a, b) = a \cdot b / (\|a\| \|b\|) \cdot = a \cdot b \text{ when } \|a\| = \|b\| = 1$$

```
from sentence_transformers import SentenceTransformer

model = SentenceTransformer("all-MiniLM-L6-v2") # 384 dimensions
docs = ["Reset your password from Settings", "Refunds take 5 to 7 days"]
D = model.encode(docs, normalize_embeddings=True)
q = model.encode("how do I change my password", normalize_embeddings=True)
print(docs[(D @ q).argmax()]) # Reset your password from Settings
```

sentence-transformers • the library from the Sentence-BERT authors • [sbert.net](https://www.sbert.net)

Found this useful?

Follow SKD Neuron for more explainers on RAG, AI agents and AI system design. Save this for your next interview.

@SKDNEURON
SKDNEURON.COM