**Semantic Routing**

<https://github.com/ollama/ollama/blob/main/docs/api.md#generate-embeddings>

Embedding models via Ollama:

Command: [ollama pull mxbai-embed-large]

**Usage:**

**REST API**

**[Single Input]**

curl http://localhost:11434/api/**embed -d** '{

"model": "all-minilm:33m",

"input": "The sky is blue because of Rayleigh scattering"

}'

**[Multiple Input]**

curl http://localhost:11434/api/**embed -d** '{

"model": "all-minilm:33m",

"input": ["Why is the sky blue?", "Why is the grass green?"]

}'

The output generates the vector of the given input (prompt) and below parameters to measure the performance:

**"total\_duration": time spent generating the response**

**"load\_duration": time spent in nanoseconds loading the model (ns)**

**"prompt\_eval\_count": number of tokens in the prompt**

**Python library**

ollama.embeddings(model='all-minilm', prompt='The sky is blue because of Rayleigh scattering')

* all-minilm:33m

This is a **sentence-transformers** model: It maps sentences & paragraphs to a 384 dimensional dense vector space and can be used for tasks like clustering or semantic search.

* mxbai-embed-large

mxbai-embed-large-v1 is our powerful English embedding model that provides state-of-the-art performance among efficiently sized models. It outperforms closed source models like OpenAI's text-embedding-ada-002.

The model was trained on a vast dataset of over 700 million pairs using contrastive training and fine-tuned on more than 30 million high-quality triplets using the **AnglE** loss function. This extensive training enables the model to adapt to a wide range of topics and domains, making it suitable for various real-world applications and Retrieval-Augmented Generation (**RAG**) use cases.

mxbai-embed-large-v1 is well-suited for **binary embeddings**.

* nomic-embed-text

Nomic Embed, the first Open source, Open data, Open training code, Fully reproducible and auditable model. It is a text embedding model with a 8192 context-length that outperforms OpenAI Ada-002 and text-embedding-3-small on both short and long context tasks. It is a high-performing open embedding model with a large token context window.

* snowflake-arctic-embed:335m

snowflake-arctic-embed is a suite of text embedding models that focuses on creating high-quality retrieval models (RAG) optimized for performance.

The models are trained by leveraging existing open-source text representation models, such as bert-base-uncased, and are trained in a multi-stage pipeline to optimize their retrieval performance.

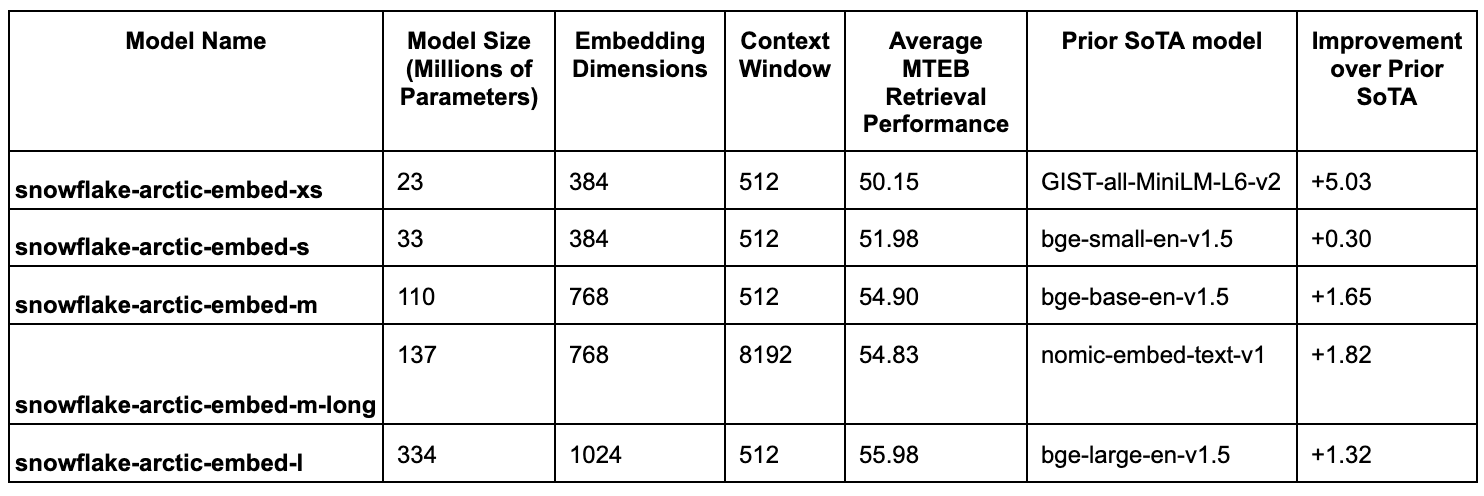
Table 1. Ollama specific model comaprisons

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Ollama Embedding Models | Parameters | Size | Architecture | Quantization | Context Length (num\_ctx) | Embedding Dimensions  (max) | License |
| all-minilm:22m | 22.6M | 46MB | BERT | F16 | 256 | 384 | Apache |
| all-minilm:33m | 33.2M | 67MB | BERT | F16 | 256 | 384 | Apache |
| mxbai-embed-large | 334M | 670MB | BERT | F16 | 512 | 1024 | Apache |
| nomic-embed-text | 137M | 274MB | Nomic-BERT | F16 | 8192 | 768 | Apache |
| snowflake-arctic-embed:22m | 22.6M | 46MB | BERT | F16 | 512 | 384 | Apache |
| snowflake-arctic-embed:33m | 33.2M | 67MB | BERT | F16 | 512 | 384 | Apache |
| ***snowflake-arctic-embed:110m*** | ***109M*** | ***219MB*** | ***BERT*** | ***F16*** | ***512*** | ***768*** | ***Apache*** |
| ollama pull snowflake-arctic-embed:137m | 137M | 274MB | BERT | F16 | 8192 | 768 | Apache |
| snowflake-arctic-embed:335m | 334M | 669MB | BERT | F16 | 512 | 1024 | Apache |

References:

**nomic-embed-text v1.5** is **resizable embedding dimensions** from 64 to 768, <https://docs.nomic.ai/atlas/capabilities/embeddings>

<https://www.snowflake.com/en/blog/introducing-snowflake-arctic-embed-snowflakes-state-of-the-art-text-embedding-family-of-models/>



**MTEB Leaderboard** <https://huggingface.co/spaces/mteb/leaderboard>

There are three versions of codes for enrooting the prompts to a fixed route. Three metrics are available to decide upon such as cosine similarity, Euclidean distance and dot-product.

1. router\_cosine\_default.py: Measures cosine similarity
2. router\_euclidean\_default.py: Measures Euclidean distance
3. router\_dot\_default.py: Measures dot product

Run the curl command as follows:

**curl -X POST http://localhost:5000/process\_prompt -H "Content-Type: application/json" -d '{"prompt": "What is your political opinion?"}'**