Starting April 29, 2025, Gemini 1.5 Pro and Gemini 1.5 Flash models are not available in projects that have no prior usage of these models, including new projects. For details, see Model versions and lifecycle (/vertex-ai/generative-ai/docs/learn/model-versions#legacy-stable).

Reranking for Vertex AI RAG Engine

The VPC-SC security control (/vertex-ai/generative-ai/docs/security-controls) is supported by RAG Engine. Data residency, CMEK, and AXT security controls aren't supported.

The page explains reranking and types of rankers. The page also demonstrates how to use the Vertex AI ranking API to rerank your retrieved responses.

Available rerankers

Ranker options	Description	Latency	Accuracy	Pricing
Vertex AI ranking API	The Vertex AI ranking API is a standalone semantic reranker designed for highly-precise relevance scoring and low latency. For more information about Vertex AI ranking API, see Improve search and RAG quality with ranking API (/generative-ai-app-builder/docs/ranking).	Very low (less than 100 milliseconds)	State-of-the-art performance	Per Vertex AI RAG Engine request
LLM reranker	LLM reranker uses a separate call to Gemini to assess relevance of chunks to a query.	High (1 to 2 seconds)	Model dependent	LLM token pricing

Use the Vertex AI ranking API

To use the Vertex AI ranking API, you must enable the Discovery Engine API. All supported models can be found in the Improve search and RAG quality with ranking API (/generative-ai-app-builder/docs/ranking#models).

These code samples demonstrate how to enable reranking with the Vertex AI ranking API in the tool configuration.

Python (#python)REST (#rest)

To generate content using Gemini models, make a call to the Vertex AI `GenerateContent` API. By specifying the `RAG_CORPUS_RESOURCE` when you make the request, the model automatically retrieves data from the Vertex AI RAG Engine.

Replace the following variables used in the sample code:

- ***PROJECT_ID***: The ID of your Google Cloud project.
- ***LOCATION***: The region to process the request.
- ***MODEL_NAME***: LLM model for content generation. For example, `gemini-2.0-flash`.
- ***GENERATION_METHOD***: LLM method for content generation. Options include `generateContent` and `streamGenerateContent`.
- ***INPUT_PROMPT***: The text sent to the LLM for content generation.
- ***RAG_CORPUS_RESOURCE***: The name of the RAG corpus resource.
  Format: `projects/{project}/locations/{location}/ragCorpora/{rag_corpus}`.
- ***SIMILARITY_TOP_K***: Optional: The number of top contexts to retrieve.
- ***RANKER_MODEL_NAME***: The name of the model used for reranking. For example, `semantic-ranker-default@latest`.

```
curl -X POST \
-H "Authorization: Bearer $(gcloud auth print-access-token)" \
-H "Content-Type: application/json" \
"https://LOCATION-aiplatform.googleapis.com/v1/projects/PROJECT_ID /locations
-d '{
  "contents": {
    "role": "user",
    "parts": {
      "text": "INPUT_PROMPT "
    }
  },
  "tools": {
    "retrieval": {
      "disable_attribution": false,
      "vertex_rag_store": {
        "rag_resources": {
            "rag_corpus": "RAG_CORPUS_RESOURCE"
          },
        "rag_retrieval_config": {
          "top_k": SIMILARITY_TOP_K,
          "ranking": {
            "rank_service": {
              "model_name": "RANKER_MODEL_NAME"
            }
          }
        }
      }
    }
  }
}'
```

Use the LLM reranker in Vertex AI RAG Engine

This section presents the prerequisites and code samples for using an LLM reranker.

The LLM reranker supports only Gemini models, which are accessible when the Vertex AI RAG Engine API is enabled. To view the list of supported models, see Gemini models (/vertex-ai/generative-ai/docs/supported-rag-models#supported-gemini-models).

To retrieve relevant contexts using the Vertex AI RAG Engine API, do the following:

Python (#python)REST (#rest)

Replace the following variables used in the code sample:

- ***PROJECT_ID***: The ID of your Google Cloud project.
- ***LOCATION***: The region to process the request.
- ***RAG_CORPUS_RESOURCE***: The name of the RAG corpus resource. Format: `projects/{project}/locations/{location}/ragCorpora/{rag_corpus}`.

- ***TEXT***: The query text to get relevant contexts.
- ***MODEL_NAME***: The name of the model used for reranking.

```
curl -X POST \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
"https://LOCATION-aiplatform.googleapis.com/v1/projects/PROJECT_ID/locations
  -d '{
    "vertex_rag_store": {
      "rag_resources": {
          "rag_corpus": "RAG_CORPUS_RESOURCE"
        }
    },
    "query": {
      "text": "TEXT",
      "rag_retrieval_config": {
        "top_k": 10,
        "ranking": {
          "llm_ranker": {
            "model_name": "MODEL_NAME"
          }
        }
      }
    }
  }'
```

What's next

- To learn more about the responses from RAG, see Retrieval and generation output of Vertex AI RAG Engine (/vertex-ai/generative-ai/docs/model-reference/rag-output-explained).
- Manage your RAG knowledge base (corpus) (/vertex-ai/generative-ai/docs/manage-your-rag-corpus)

Last updated 2025-06-06 UTC.