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).

Text embeddings API

Release Notes

The Text embeddings API converts textual data into numerical vectors. These vector representations are designed to capture the semantic meaning and context of the words they represent.

Supported Models:

You can get text embeddings by using the following models:

Model name	Description	Output Dimensions	Max sequence length	Supported text la
`gemini-embedding-001`	State-of-the-art performance across English, multilingual and code tasks. It unifies the previously specialized models like `text-embedding-005` and `text-multilingual-embedding-002` and achieves better performance in their respective domains. Read our Tech Report (https://deepmind.google/research/publications/157741/) for more detail.	up to 3072	2048 tokens	Supported text la (/vertex-ai/gene ai/docs/model-r embeddings-api#supported_t
`text-embedding-005`	Specialized in English and code tasks.	up to 768	2048 tokens	English
`text-multilingual-embedding-002`	Specialized in multilingual tasks.	up to 768	2048 tokens	Supported text la (/vertex-ai/gene ai/docs/model-r embeddings-api#supported_t

For superior embedding quality, `gemini-embedding-001` is our large model designed to provide the highest performance. Note that `gemini-embedding-001` supports one instance per request.

Syntax

curlPython (#python)
(#curl)

```
PROJECT_ID = PROJECT_ID
REGION = us-central1
MODEL_ID = MODEL_ID

curl -X POST \
  -H "Authorization: Bearer $(gcloud auth print-access-token)" \
  -H "Content-Type: application/json" \
  https://${REGION}-aiplatform.googleapis.com/v1/projects/${PROJECT_ID}/locatio
  '{
    "instances": [
      ...
    ],
    "parameters": {
      ...
    }
  }'
```

Parameter list

Top-level fields	
`instances`	A list of objects containing the following fields: • `content` • `title` (optional) • `task_type` (optional)
`parameters`	An object containing the following fields: • `autoTruncate` (optional) • `outputDimensionality` (optional)

`instance` fields

`content`	`string` The text that you want to generate embeddings for.
`task_type`	Optional: `string` Used to convey intended downstream application to help the model produce better embeddings. If left blank, the default used is `RETRIEVAL_QUERY`. • `RETRIEVAL_QUERY` • `RETRIEVAL_DOCUMENT` • `SEMANTIC_SIMILARITY` • `CLASSIFICATION` • `CLUSTERING` • `QUESTION_ANSWERING` • `FACT_VERIFICATION` • `CODE_RETRIEVAL_QUERY` For more information about task types, see Choose an embeddings task type (/vertex-ai/generative-ai/docs/embeddings/task-types).
`title`	Optional: `string` Used to help the model produce better embeddings. Only valid with `task_type=RETRIEVAL_DOCUMENT`.

`task_type`

The following table describes the `task_type` parameter values and their use cases:

`task_type`	Description
`RETRIEVAL_QUERY`	Specifies the given text is a query in a search or retrieval setting. Use RETRIEVAL_DOCUMENT for the document side.
`RETRIEVAL_DOCUMENT`	Specifies the given text is a document in a search or retrieval setting.
`SEMANTIC_SIMILARITY`	Specifies the given text is used for Semantic Textual Similarity (STS).
`CLASSIFICATION`	Specifies that the embedding is used for classification.

`task_type`	Description
`CLUSTERING`	Specifies that the embedding is used for clustering.
`QUESTION_ANSWERING`	Specifies that the query embedding is used for answering questions. Use RETRIEVAL_DOCUMENT for the document side.
`FACT_VERIFICATION`	Specifies that the query embedding is used for fact verification. Use RETRIEVAL_DOCUMENT for the document side.
`CODE_RETRIEVAL_QUERY`	Specifies that the query embedding is used for code retrieval for Java and Python. Use RETRIEVAL_DOCUMENT for the document side.

Retrieval Tasks:

Query: Use task_type=`RETRIEVAL_QUERY` to indicate that the input text is a search query. Corpus: Use task_type=`RETRIEVAL_DOCUMENT` to indicate that the input text is part of the document collection being searched.

Similarity Tasks:

Semantic similarity: Use task_type= `SEMANTIC_SIMILARITY` for both input texts to assess their overall meaning similarity.

Note: `SEMANTIC_SIMILARITY` is not intended for retrieval use cases, such as document search and information retrieval. For these use cases, use **`RETRIEVAL_DOCUMENT`**, **`RETRIEVAL_QUERY`**, **`QUESTION_ANSWERING`**, and **`FACT_VERIFICATION`**.

`parameters` fields	
`autoTruncate`	Optional: `bool` When set to true, input text will be truncated. When set to false, an error is returned if the input text is longer than the maximum length supported by the model. Defaults to true.
`outputDimensionality`	Optional: `int` Used to specify output embedding size. If set, output embeddings will be truncated to the size specified.

Request body

```
{
  "instances": [
    {
      "task_type": "RETRIEVAL_DOCUMENT",
      "title": "document title",
      "content": "I would like embeddings for this text!"
    },
  ]
}
```

Response body

```
{
  "predictions": [
    {
      "embeddings": {
        "statistics": {
          "truncated": boolean,
          "token_count": integer
        },
        "values": [ number ]
      }
    }
  ]
}
```

Response elements

`predictions`

A list of objects with the following fields:

- **`embeddings`**: The result generated from input text. Contains the following fields:
    - **`values`**
    - **`statistics`**

`embeddings` fields

`values`	A list of `float`s. The `values` field contains a numerical encoding (embedding vector) of the semantic content present in the given input text.
`statistics`	The statistics computed from the input text. Contains: - `truncated`: Indicates whether the input text was truncated due to being longer than the maximum number of tokens allowed by the model. - `token_count`: Number of tokens of the input text.

Sample response

```
{
  "predictions": [
    {
      "embeddings": {
        "values": [
          0.0058424929156899452,
          0.011848051100969315,
          0.032247550785541534,
          -0.031829461455345154,
          -0.055369812995195389,
          ...
        ],
        "statistics": {
          "token_count": 4,
          "truncated": false
        }
      }
    }
  ]
}
```

Examples

Embed a text string

The following example shows how to obtain the embedding of a text string.

REST (#rest) Vertex AI SDK for Python... Go (#go) Java (#java) Node.js (#node.js)

After you set up your environment (/vertex-ai/generative-ai/docs/start/quickstarts/quickstart-multimodal#gemini-setup-environment-drest), you can use REST to test a text prompt. The following sample sends a request to the publisher model endpoint.

Before using any of the request data, make the following replacements:

- *PROJECT_ID*: Your project ID (/resource-manager/docs/creating-managing-projects#identifiers).
- *TEXT*: The text that you want to generate embeddings for. **Limit:** five texts of up to 2,048 tokens per text for all models except `textembedding-gecko@001`. The max input token length for `textembedding-gecko@001` is 3072. For `gemini-embedding-001`, each request can only include a single input text. For more information, see Text embedding limits (/vertex-ai/docs/quotas#text-embedding-limits).
- *AUTO_TRUNCATE*: If set to `false`, text that exceeds the token limit causes the request to fail. The default value is `true`.

HTTP method and URL:

```
POST https://us-central1-aiplatform.googleapis.com/v1/projects/PROJECT_ID/lo
```

Request JSON body:

```
{
  "instances": [
    { "content": "TEXT"}
  ],
  "parameters": {
    "autoTruncate": AUTO_TRUNCATE
  }
}
```

To send your request, choose one of these options:

curlPowerShell (#powershell)
(#curl)

★ **Note:** The following command assumes that you have logged in to the **`gcloud`** CLI with your user account by running **`gcloud init`** (/sdk/gcloud/reference/init) or **`gcloud auth login`** (/sdk/gcloud/reference/auth/login) , or by using Cloud Shell (/shell/docs), which automatically logs you into the **`gcloud`** CLI . You can check the currently active account by running **`gcloud auth list`** (/sdk/gcloud/reference/auth/list).

Save the request body in a file named **`request.json`**, and execute the following command:

```
curl -X POST \
     -H "Authorization: Bearer $(gcloud auth print-access-token)" \
     -H "Content-Type: application/json; charset=utf-8" \
     -d @request.json \
     "https://us-central1-aiplatform.googleapis.com/v1/projects/PROJECT_I
```

You should receive a JSON response similar to the following. Note that **`values`** has been truncated to save space.

⊕ **Response**

```
{
  "predictions": [
    {
      "embeddings": {
        "statistics": {
          "truncated": false,
          "token_count": 6
        },
        "values": [ ... ]
      }
    }
  ]
}
```

Note the following in the URL for this sample:

- Use the `generateContent` (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/generateContent) method to request that the response is returned after it's fully generated. To reduce the perception of latency to a human audience, stream the response as it's being generated by using the `streamGenerateContent` (/vertex-ai/docs/reference/rest/v1/projects.locations.publishers.models/streamGenerateContent) method.
- The multimodal model ID is located at the end of the URL before the method (for example, `gemini-2.0-flash`). This sample might support other models as well.

Supported text languages

All text embedding models support and have been evaluated on English-language text. The `text-multilingual-embedding-002` model additionally supports and has been evaluated on the following languages:

- **Evaluated languages:** `Arabic (ar)`, `Bengali (bn)`, `English (en)`, `Spanish (es)`, `German (de)`, `Persian (fa)`, `Finnish (fi)`, `French (fr)`, `Hindi (hi)`, `Indonesian (id)`, `Japanese (ja)`, `Korean (ko)`, `Russian (ru)`, `Swahili (sw)`, `Telugu (te)`, `Thai (th)`, `Yoruba (yo)`, `Chinese (zh)`
- **Supported languages**: `Afrikaans`, `Albanian`, `Amharic`, `Arabic`, `Armenian`, `Azerbaijani`, `Basque`, `Belarusiasn`, `Bengali`, `Bulgarian`, `Burmese`, `Catalan`, `Cebuano`, `Chichewa`, `Chinese`, `Corsican`, `Czech`, `Danish`, `Dutch`, `English`, `Esperanto`, `Estonian`, `Filipino`, `Finnish`, `French`, `Galician`, `Georgian`, `German`, `Greek`, `Gujarati`, `Haitian Creole`, `Hausa`, `Hawaiian`, `Hebrew`, `Hindi`, `Hmong`, `Hungarian`, `Icelandic`, `Igbo`, `Indonesian`, `Irish`, `Italian`, `Japanese`, `Javanese`, `Kannada`, `Kazakh`, `Khmer`, `Korean`, `Kurdish`, `Kyrgyz`, `Lao`, `Latin`, `Latvian`, `Lithuanian`, `Luxembourgish`, `Macedonian`, `Malagasy`, `Malay`, `Malayalam`, `Maltese`, `Maori`, `Marathi`, `Mongolian`, `Nepali`, `Norwegian`, `Pashto`, `Persian`, `Polish`, `Portuguese`, `Punjabi`, `Romanian`, `Russian`, `Samoan`, `Scottish Gaelic`, `Serbian`, `Shona`, `Sindhi`, `Sinhala`, `Slovak`, `Slovenian`, `Somali`, `Sotho`, `Spanish`, `Sundanese`, `Swahili`, `Swedish`, `Tajik`, `Tamil`, `Telugu`, `Thai`, `Turkish`, `Ukrainian`, `Urdu`, `Uzbek`, `Vietnamese`, `Welsh`, `West Frisian`, `Xhosa`, `Yiddish`, `Yoruba`, `Zulu`.

The `gemini-embedding-001` model supports the following languages:

`Arabic`, `Bengali`, `Bulgarian`, `Chinese (Simplified and Traditional)`, `Croatian`, `Czech`, `Danish`, `Dutch`, `English`, `Estonian`, `Finnish`, `French`, `German`, `Greek`, `Hebrew`, `Hindi`, `Hungarian`, `Indonesian`, `Italian`, `Japanese`, `Korean`, `Latvian`, `Lithuanian`, `Norwegian`, `Polish`, `Portuguese`, `Romanian`, `Russian`, `Serbian`, `Slovak`, `Slovenian`, `Spanish`, `Swahili`, `Swedish`, `Thai`, `Turkish`, `Ukrainian`, `Vietnamese`, `Afrikaans`, `Amharic`, `Assamese`, `Azerbaijani`, `Belarusian`, `Bosnian`, `Catalan`, `Cebuano`, `Corsican`, `Welsh`, `Dhivehi`, `Esperanto`, `Basque`, `Persian`, `Filipino (Tagalog)`, `Frisian`, `Irish`, `Scots Gaelic`, `Galician`, `Gujarati`, `Hausa`, `Hawaiian`, `Hmong`, `Haitian Creole`, `Armenian`, `Igbo`, `Icelandic`, `Javanese`, `Georgian`, `Kazakh`, `Khmer`, `Kannada`, `Krio`, `Kurdish`, `Kyrgyz`, `Latin`, `Luxembourgish`, `Lao`, `Malagasy`, `Maori`, `Macedonian`, `Malayalam`, `Mongolian`, `Meiteilon (Manipuri)`, `Marathi`, `Malay`, `Maltese`, `Myanmar (Burmese)`, `Nepali`, `Nyanja (Chichewa)`, `Odia (Oriya)`, `Punjabi`, `Pashto`, `Sindhi`, `Sinhala (Sinhalese)`, `Samoan`, `Shona`, `Somali`, `Albanian`, `Sesotho`, `Sundanese`, `Tamil`, `Telugu`, `Tajik`, `Uyghur`, `Urdu`, `Uzbek`, `Xhosa`, `Yiddish`, `Yoruba`, `Zulu`.

Model versions

To use a current stable model, specify the model version number, for example `gemini-embedding-001`. Specifying a model without a version number, isn't recommended, as it is merely a legacy pointer to another model and isn't stable.

For more information, see Model versions and lifecycle (/vertex-ai/generative-ai/docs/learn/model-versioning).

What's next

For detailed documentation, see the following:

- Text Embeddings (/vertex-ai/generative-ai/docs/embeddings/get-text-embeddings)

Last updated 2025-06-06 UTC.