Text embeddings API

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:

English models	Multilingual models	Gemini embedding models
`textembedding-gecko@001`	`textembedding-gecko-multilingual@001`	`text-embedding-large-exp-03-07` (experimental)
`textembedding-gecko@003`	`text-multilingual-embedding-002`	
`text-embedding-004`		
`text-embedding-005`		

Note: **`textembedding-gecko@001`** is being discontinued and will no longer be available after April 09, 2025. Migrate to **`text-embedding-005`** to avoid service disruptions.

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}/
  '{
    "instances": [
      ...
    ],
    "parameters": {
      ...
    }
  }'
```

Parameter list

Parameters	
`texts`	`list of union[string, TextEmbeddingInput]` Each instance represents a single piece of text to be embedded.
`TextEmbeddingInput`	`string` The text that you want to generate embeddings for.
`auto_truncate`	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.
`output_dimensionality`	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!"
    },
  ]
}
```

Parameters	
`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` The `task_type` parameter is not supported for the textembedding-gecko@001 model. 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`.

taskType

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.
`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).

`task_type`	Description
`CLASSIFICATION`	Specifies that the embedding is used for classification.
`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.
`CODE_RETRIEVAL_QUERY`	Specifies that the query embedding is used for code retrieval for Java and Python.

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.

Response body

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

Response element	Description
`embeddings`	The result generated from input text.
`statistics`	The statistics computed from the input text.
`truncated`	Indicates if the input text was longer than max allowed tokens and truncated.
`tokenCount`	Number of tokens of the input text.
`values`	The `values` field contains the embedding vectors corresponding to the words in 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

Basic use case

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

RESTVertex AI SDK for Python... (#rest) 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.
- *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_I
```

Request JSON body:

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

To send your request, choose one of these options:

curl (#curl)PowerShell

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) . 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:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }

Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://us-central1-aiplatform.googleapis.com/v1/project
```

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-1.5-flash` or `gemini-1.0-pro-vision`). This sample may support other models as well.

Advanced Use Case

The following example demonstrates some advanced features

- Use `task_type` and `title` to improve embedding quality.
- Use parameters to control the behavior of the API.

Note: Feature support varies by model version. For example, `task_type` and `title` fields are **not** supported by `textembedding-gecko@001`. For best results, choose the latest available version.

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

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 3,072 tokens per text.
- *TASK_TYPE*: Used to convey the intended downstream application to help the model produce better embeddings.
- *TITLE*: Used to help the model produce better embeddings.
- *AUTO_TRUNCATE*: If set to `false`, text that exceeds the token limit causes the request to fail. The default value is `true`.

- *OUTPUT_DIMENSIONALITY*: Used to specify output embedding size. If set, output embeddings will be truncated to the size specified.

HTTP method and URL:

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

Request JSON body:

```
{
  "instances": [
    { "content": "TEXT",
      "task_type": "TASK_TYPE",
      "title": "TITLE"
    },
  ],
  "parameters": {
    "autoTruncate": AUTO_TRUNCATE,
    "outputDimensionality": OUTPUT_DIMENSIONALITY
  }
}
```

To send your request, choose one of these options:

curl (#curl)PowerShell (#powershell)

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) . 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:

```
$cred = gcloud auth print-access-token
$headers = @{ "Authorization" = "Bearer $cred" }
```

```
Invoke-WebRequest `
    -Method POST `
    -Headers $headers `
    -ContentType: "application/json; charset=utf-8" `
    -InFile request.json `
    -Uri "https://us-central1-aiplatform.googleapis.com/v1/project
```

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": [ ... ]
      }
    }
  ]
}
```

Supported text languages

All text embedding models support and have been evaluated on English-language text. The `textembedding-gecko-multilingual@001` and `text-multilingual-embedding-002` models additionally support and have 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.

Model versions

To use a stable model version (/vertex-ai/generative-ai/docs/learn/model-versioning#stable-version), specify the model version number, for example `text-embedding-005`. Specifying a model without a version number, such as `textembedding-gecko`, isn't recommended, as it is merely a legacy pointer to another model and isn't stable. Each stable version is available for six months after the release date of the subsequent stable version.

The following table contains the available stable model versions:

Model name	Release date	Discontinuation date
text-embedding-005	Nov 18, 2024	To be determined.
text-embedding-004	May 14, 2024	Nov 18, 2025
text-multilingual-embedding-002	May 14, 2024	To be determined.
textembedding-gecko@003	December 12, 2023	May 24, 2025
textembedding-gecko-multilingual@001	November 2, 2023	May 24, 2025
textembedding-gecko@002 (regressed, but still supported)	November 2, 2023	April 9, 2025
textembedding-gecko@001 (regressed, but still supported)	June 7, 2023	April 9, 2025

Model name	Release date	Discontinuation date
multimodalembedding@001	February 12, 2024	To be determined.

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 2025-03-21 UTC.