Package 'PaLMr'

July 21, 2025

Type Package

Title Interface for 'Google Pathways Language Model 2 (PaLM 2)'

Version 0.2.0

Maintainer Li Yuan <lyuan@gd.edu.kg>

Description 'Google Pathways Language Model 2 (PaLM 2)' as a coding and writing assistant designed for 'R'. With a range of functions, including natural language processing and coding optimization, to assist 'R' developers in simplifying tedious coding tasks and content searching.

License CC BY 4.0

URL https://palmr.ly.gd.edu.kg/

BugReports https://github.com/lygitdata/PaLMr/issues

Encoding UTF-8

RoxygenNote 7.3.0

Imports httr, jsonlite

NeedsCompilation no

Author Li Yuan [aut, cre] (ORCID: <https://orcid.org/0009-0008-1075-9922>)

Repository CRAN

Date/Publication 2024-01-22 04:42:44 UTC

Contents

`palm.connect` *Set up and connect to the Google PaLM 2 text model.*

Description

This function establishes a connection to the Google PaLM text model by specifying the API key, and model version.

Usage

```
palm.connect(version, api, proxy = FALSE)
```

Arguments

`version`	A character string representing the model version to use, either `"v1beta2"` or `"v1beta3"`.
`api`	A character string representing the API key for accessing the Google PaLM 2 text model. The API key should be 39 characters long and must be of the "character" class.
`proxy`	A boolean value indicating whether to use a proxy for accessing the API URL (default is `FALSE`). If your local internet cannot access the APIs, set this parameter to `TRUE`.

Details

This function performs the necessary setup to connect to the Google PaLM 2 text model. It validates the provided API key and checks the correctness of the model version. If the input is valid, it constructs the API request and sends it to the PaLM 2 API endpoint.

If an error occurs during the API request, such as an invalid API key or input parameters, an error message is displayed. If the API request is successful, the function prints the model details to the console and returns a character vector with the API key, model version, and model type.

Value

If successful, the function returns a character vector containing the API key, model version, and proxy status. If the API response indicates an error, the function stops execution and provides an error message.

See Also

PaLMr - Documentation

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)
palm.model

## End(Not run)
```

---

palm.txt *Generate text using the Google PaLM 2 text model based on a prompt*

Description

This function sends a prompt to the Google PaLM 2 text model and generates text as a response. It allows customization of the generated text using various parameters.

Usage

```
palm.txt(
  model.parameter,
  prompt,
  temperature = 0.7,
  maxOutputTokens = 1024,
  topP = 0.95,
  topK = 40,
  htUnspecified = "meda",
  htDerogatory = "meda",
  htToxicity = "meda",
  htViolence = "meda",
  htSexual = "meda",
  htMedical = "meda",
  htDangerous = "meda"
)
```

Arguments

model.parameter	A character vector containing the API key, model version, and proxy status. Model version and type are specified by Google. See function palm.connect for detail.
prompt	A character string representing the query or prompt for text generation. The length of the query should be between 1 and 8196 characters, inclusive.

`temperature` A numeric value between 0.0 and 1.0, inclusive (default: 0.7). Controls the randomness of the generated text. A higher value (e.g., 0.9) results in more creative responses, while a lower value (e.g., 0.3) produces more straightforward text.

`maxOutputTokens`
An integer value (default: 1024). Specifies the maximum number of tokens to include in the generated text.

`topP` A numeric value (default: 0.95). Defines the maximum cumulative probability of tokens considered when sampling. It controls the diversity of the text generated.

`topK` An integer value (default: 40). Sets the maximum number of tokens to consider when sampling.

`htUnspecified` Safety setting threshold for unspecified harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDerogatory` Safety setting threshold for derogatory harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htToxicity` Safety setting threshold for toxicity harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htViolence` Safety setting threshold for violence harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htSexual` Safety setting threshold for sexual harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htMedical` Safety setting threshold for medical harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDangerous` Safety setting threshold for dangerous harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

Details

This function interacts with the Google PaLM model by sending a query using the specified parameters. It allows you to customize the generated text by adjusting the ‘temperature‘, ‘maxOutputTokens‘, ‘topP‘, ‘topK‘, and safety settings.

If the function is successful, it returns a character string containing the generated text. If an error occurs during the API request, it will stop execution and provide an error message.

The ‘model.parameter‘ argument should be a character vector with the API key, model version, and model type provided by Google. You can obtain this information by following the instructions provided by Google for using the PaLM API.

The safety settings control the content’s safety level based on different harm categories. Harm thresholds are specified as per Google’s guidelines and can be customized to control the content generated.

Value

A character string generated by the Google PaLM 2 API based on the provided prompt and parameters.

See Also

PaLMr - Documentation

Safety Setting - Google AI for Developers

HarmCategory - Google AI for Developers

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)

prompt = "Write a story about a magic backpack."
generated.text = palm.txt(palm.model,
                          prompt)
cat(generated.text)

## End(Not run)
```

`palm.txt.explain.code` *Explain code based on a query using the Google PaLM 2 text model*

Description

This function sends a query with a code snippet to the Google PaLM 2 text model and generates a detailed explanation of the code. It supports various programming languages and allows you to customize the explanation.

Usage

```
palm.txt.explain.code(
  model.parameter,
  prompt,
  language = "R",
  temperature = 0.7,
  maxOutputTokens = 1024,
  topP = 0.95,
  topK = 40,
  htUnspecified = "meda",
  htDerogatory = "meda",
  htToxicity = "meda",
  htViolence = "meda",
  htSexual = "meda",
  htMedical = "meda",
  htDangerous = "meda"
)
```

Arguments

`model.parameter`
: A character vector containing the API key, model version, and proxy status. Model version and type are specified by Google. See function `palm.connect` for detail.

`prompt` A character string representing the code snippet for explanation. The length of the code snippet should be between 1 and 8196 characters, inclusive.

`language` A character string specifying the programming language used in the code (default: "R").

`temperature` A numeric value between 0.0 and 1.0, inclusive (default: 0.7). Controls the randomness of the generated explanation. A higher value (e.g., 0.9) results in more creative responses, while a lower value (e.g., 0.3) produces more straightforward explanations.

`maxOutputTokens`
An integer value (default: 1024). Specifies the maximum number of tokens to include in the generated explanation.

`topP` A numeric value (default: 0.95). Defines the maximum cumulative probability of tokens considered when sampling. It controls the diversity of the explanation generated.

`topK` An integer value (default: 40). Sets the maximum number of tokens to consider when sampling.

`htUnspecified` Safety setting threshold for unspecified harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDerogatory` Safety setting threshold for derogatory harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htToxicity` Safety setting threshold for toxicity harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htViolence` Safety setting threshold for violence harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE

"high" BLOCK_ONLY_HIGH

"none" BLOCK_NONE

`htSexual` Safety setting threshold for sexual harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED

"lowa" BLOCK_LOW_AND_ABOVE

"meda" BLOCK_MEDIUM_AND_ABOVE

"high" BLOCK_ONLY_HIGH

"none" BLOCK_NONE

`htMedical` Safety setting threshold for medical harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED

"lowa" BLOCK_LOW_AND_ABOVE

"meda" BLOCK_MEDIUM_AND_ABOVE

"high" BLOCK_ONLY_HIGH

"none" BLOCK_NONE

`htDangerous` Safety setting threshold for dangerous harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED

"lowa" BLOCK_LOW_AND_ABOVE

"meda" BLOCK_MEDIUM_AND_ABOVE

"high" BLOCK_ONLY_HIGH

"none" BLOCK_NONE

Details

This function interacts with the Google PaLM model by sending a code query to explain code. It allows you to customize the generated code explanations by specifying the programming language, and additional parameters like temperature, token limits, and safety settings.

If the function is successful, it returns a detailed explanation of the provided code as a character string. If an error occurs during the API request, it will stop execution and provide an error message.

The ‘model.parameter‘ argument should be a character vector with the API key, model version, and model type provided by Google. You can obtain this information by following the instructions provided by Google for using the PaLM API.

The safety settings control the content's safety level based on different harm categories. Harm thresholds are specified as per Google's guidelines and can be customized to control the content generated.

Value

A character string containing the detailed explanation of the provided code snippet based on the query and parameters.

See Also

PaLMr - Documentation

Safety Setting - Google AI for Developers

HarmCategory - Google AI for Developers

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)

prompt = "foo <- function(n) {
            if (n <= 0) {
              return(0)
            } else if (n == 1) {
              return(1)
            } else {
              return(foo(n - 1) + foo(n - 2))
            }
          }"
code.explanation = palm.txt.explain.code(palm.model,
                                         prompt)
cat(code.explanation)

## End(Not run)
```

`palm.txt.fix.grammar` *Fix grammar and rewrite text using the Google PaLM 2 text model*

Description

This function sends a query with grammatical issues to the Google PaLM 2 text model and generates corrected text as a response. It allows customization of the generated text using various parameters.

Usage

```
palm.txt.fix.grammar(
  model.parameter,
  prompt,
  temperature = 0.7,
  maxOutputTokens = 1024,
  topP = 0.95,
  topK = 40,
  htUnspecified = "meda",
  htDerogatory = "meda",
```

```
  htToxicity = "meda",
  htViolence = "meda",
  htSexual = "meda",
  htMedical = "meda",
  htDangerous = "meda"
)
```

Arguments

model.parameter	A character vector containing the API key, model version, and proxy status. Model version and type are specified by Google. See function palm.connect for detail.
prompt	A character string representing the code snippet for explanation. The length of the code snippet should be between 1 and 8196 characters, inclusive.
temperature	A numeric value between 0.0 and 1.0, inclusive (default: 0.7). Controls the randomness of the generated text. A higher value (e.g., 0.9) results in more creative responses, while a lower value (e.g., 0.3) produces more straightforward text.
maxOutputTokens	An integer value (default: 1024). Specifies the maximum number of tokens to include in the generated text.
topP	A numeric value (default: 0.95). Defines the maximum cumulative probability of tokens considered when sampling. It controls the diversity of the text generated.
topK	An integer value (default: 40). Sets the maximum number of tokens to consider when sampling.
htUnspecified	Safety setting threshold for unspecified harm. The default threshold is "meda". Valid options are as follows. **"unsp"** HARM_BLOCK_THRESHOLD_UNSPECIFIED **"lowa"** BLOCK_LOW_AND_ABOVE **"meda"** BLOCK_MEDIUM_AND_ABOVE **"high"** BLOCK_ONLY_HIGH **"none"** BLOCK_NONE
htDerogatory	Safety setting threshold for derogatory harm. The default threshold is "meda". Valid options are as follows. **"unsp"** HARM_BLOCK_THRESHOLD_UNSPECIFIED **"lowa"** BLOCK_LOW_AND_ABOVE **"meda"** BLOCK_MEDIUM_AND_ABOVE **"high"** BLOCK_ONLY_HIGH **"none"** BLOCK_NONE
htToxicity	Safety setting threshold for toxicity harm. The default threshold is "meda". Valid options are as follows. **"unsp"** HARM_BLOCK_THRESHOLD_UNSPECIFIED

"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htViolence` Safety setting threshold for violence harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htSexual` Safety setting threshold for sexual harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htMedical` Safety setting threshold for medical harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDangerous` Safety setting threshold for dangerous harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

Details

This function interacts with the Google PaLM model by sending a query with grammatical issues using the specified parameters. It allows you to customize the generated text by adjusting the ‘temperature‘, ‘maxOutputTokens‘, ‘topP‘, ‘topK‘, and safety settings.

If the function is successful, it returns a character string containing the rewritten text with corrected grammar. If an error occurs during the API request, it will stop execution and provide an error message.

The 'model.parameter' argument should be a character vector with the API key, model version, and model type provided by Google. You can obtain this information by following the instructions provided by Google for using the PaLM API.

The safety settings control the content's safety level based on different harm categories. Harm thresholds are specified as per Google's guidelines and can be customized to control the content generated.

Value

A character string containing the rewritten text with corrected grammar, generated by the Google PaLM API based on the provided query and parameters.

See Also

PaLMr - Documentation

Safety Setting - Google AI for Developers

HarmCategory - Google AI for Developers

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)

prompt = "Yesterday, I will buy a book for my younger sister as his birthday gift.
          They were very happen when seeing this gift earlier today."
correct.text = palm.txt.fix.grammar(palm.model,
                                    prompt)
cat(correct.text)

## End(Not run)
```

---

palm.txt.get.reference

Get references based on a query using the Google PaLM 2 text model

---

Description

This function sends a query to the Google PaLM 2 text model and generates a list of references based on the query. It allows customization of the generated references and supports various citation styles and source types.

Usage

```
palm.txt.get.reference(
  model.parameter,
  prompt,
  source.type = "articles",
  source.date = "most recent",
  n.source = 5,
  citation.style = "APA7",
  temperature = 0.7,
  maxOutputTokens = 1024,
  topP = 0.95,
  topK = 40,
  htUnspecified = "meda",
  htDerogatory = "meda",
  htToxicity = "meda",
  htViolence = "meda",
  htSexual = "meda",
  htMedical = "meda",
  htDangerous = "meda"
)
```

Arguments

model.parameter	A character vector containing the API key, model version, and proxy status. Model version and type are specified by Google. See function palm.connect for detail.
prompt	A character string representing the code snippet for explanation. The length of the code snippet should be between 1 and 8196 characters, inclusive.
source.type	A character string specifying the type of sources to search for (default: "articles").
source.date	A character string specifying the date range for the sources (default: "most recent").
n.source	An integer value specifying the number of sources to retrieve (default: 5).
citation.style	A character string specifying the citation style for the references (default: "APA7").
temperature	A numeric value between 0.0 and 1.0, inclusive (default: 0.7). Controls the randomness of the generated references. A higher value (e.g., 0.9) results in more creative responses, while a lower value (e.g., 0.3) produces more straightforward references.
maxOutputTokens	An integer value (default: 1024). Specifies the maximum number of tokens to include in the generated references.
topP	A numeric value between (default: 0.95). Defines the maximum cumulative probability of tokens considered when sampling. It controls the diversity of the references generated.

topK
: An integer value between (default: 40). Sets the maximum number of tokens to consider when sampling.

htUnspecified
: Safety setting threshold for unspecified harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE
  "meda" BLOCK_MEDIUM_AND_ABOVE
  "high" BLOCK_ONLY_HIGH
  "none" BLOCK_NONE

htDerogatory
: Safety setting threshold for derogatory harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE
  "meda" BLOCK_MEDIUM_AND_ABOVE
  "high" BLOCK_ONLY_HIGH
  "none" BLOCK_NONE

htToxicity
: Safety setting threshold for toxicity harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE
  "meda" BLOCK_MEDIUM_AND_ABOVE
  "high" BLOCK_ONLY_HIGH
  "none" BLOCK_NONE

htViolence
: Safety setting threshold for violence harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE
  "meda" BLOCK_MEDIUM_AND_ABOVE
  "high" BLOCK_ONLY_HIGH
  "none" BLOCK_NONE

htSexual
: Safety setting threshold for sexual harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE
  "meda" BLOCK_MEDIUM_AND_ABOVE
  "high" BLOCK_ONLY_HIGH
  "none" BLOCK_NONE

htMedical
: Safety setting threshold for medical harm. The default threshold is "meda". Valid options are as follows.

  "unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
  "lowa" BLOCK_LOW_AND_ABOVE

"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDangerous` Safety setting threshold for dangerous harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

Details

This function interacts with the Google PaLM model by sending a query to find references. It allows you to customize the generated references by specifying the number of sources, citation style, source type, date range, and safety settings.

If the function is successful, it returns a character vector containing the generated references. If an error occurs during the API request, it will stop execution and provide an error message.

The 'model.parameter' argument should be a character vector with the API key, model version, and model type provided by Google. You can obtain this information by following the instructions provided by Google for using the PaLM API.

The safety settings control the content's safety level based on different harm categories. Harm thresholds are specified as per Google's guidelines and can be customized to control the content generated.

Value

A character string containing the generated references based on the provided query and parameters.

See Also

PaLMr - Documentation

Safety Setting - Google AI for Developers

HarmCategory - Google AI for Developers

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)

correct.text = palm.txt.get.reference(palm.model,
                                      "H5N1 in the United States")
cat(correct.text)
```

```
## End(Not run)
```

---

palm.txt.optimize.code *Optimize code based on a query using the Google PaLM 2 text model.*

Description

This function sends a query with a code snippet to the Google PaLM 2 text model and generates an optimized version of the code. You can specify the programming language and the aspect you want to optimize (e.g., "runtime" or "memory"). The optimized code is provided along with the original code for comparison.

Usage

```
palm.txt.optimize.code(
  model.parameter,
  prompt,
  goal,
  language = "R",
  temperature = 0.7,
  maxOutputTokens = 1024,
  topP = 0.95,
  topK = 40,
  htUnspecified = "meda",
  htDerogatory = "meda",
  htToxicity = "meda",
  htViolence = "meda",
  htSexual = "meda",
  htMedical = "meda",
  htDangerous = "meda"
)
```

Arguments

model.parameter	A character vector containing the API key, model version, and proxy status. Model version and type are specified by Google. See function palm.connect for detail.
prompt	A character string representing the code snippet for explanation. The length of the code snippet should be between 1 and 8196 characters, inclusive.
goal	A character string specifying the aspect you want to optimize.
language	A character string specifying the programming language used in the code (default: "R").

`temperature` A numeric value (default: 0.7). Controls the randomness of the generated optimization. A higher value (e.g., 0.9) results in more creative optimizations, while a lower value (e.g., 0.3) produces more straightforward optimizations.

`maxOutputTokens` An integer value between 1 and 1024, inclusive (default: 1024). Specifies the maximum number of tokens to include in the generated optimization.

`topP` A numeric value (default: 0.95). Defines the maximum cumulative probability of tokens considered when sampling. It controls the diversity of the optimization generated.

`topK` An integer value (default: 40). Sets the maximum number of tokens to consider when sampling.

`htUnspecified` Safety setting threshold for unspecified harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDerogatory` Safety setting threshold for derogatory harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htToxicity` Safety setting threshold for toxicity harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htViolence` Safety setting threshold for violence harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htSexual` Safety setting threshold for sexual harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED

"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htMedical` Safety setting threshold for medical harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

`htDangerous` Safety setting threshold for dangerous harm. The default threshold is "meda". Valid options are as follows.

"unsp" HARM_BLOCK_THRESHOLD_UNSPECIFIED
"lowa" BLOCK_LOW_AND_ABOVE
"meda" BLOCK_MEDIUM_AND_ABOVE
"high" BLOCK_ONLY_HIGH
"none" BLOCK_NONE

Details

This function interacts with the Google PaLM model by sending a code query for code optimization. It allows you to customize the generated code optimizations by specifying the programming language, optimization aspect, and additional parameters like temperature, token limits, and safety settings.

If the function is successful, it returns an optimized version of the provided code as a character string. If an error occurs during the API request, it will stop execution and provide an error message.

The ‘model.parameter‘ argument should be a character vector with the API key, model version, and model type provided by Google. You can obtain this information by following the instructions provided by Google for using the PaLM API.

The safety settings control the content's safety level based on different harm categories. Harm thresholds are specified as per Google's guidelines and can be customized to control the content generated.

Value

A character string containing the optimized version of the provided code snippet based on the query and parameters.

See Also

PaLMr - Documentation

Safety Setting - Google AI for Developers

HarmCategory - Google AI for Developers

Examples

```
## Not run:
# Connect to the model, replace API_KEY with your api key
palm.model = palm.connect("v1beta2",
                          "API_KEY",
                          FALSE)

prompt = "foo <- function(n) {
            if (n <= 0) {
              return(0)
            } else if (n == 1) {
              return(1)
            } else {
              return(foo(n - 1) + foo(n - 2))
            }
          }"
code.optimization = palm.txt.optimize.code(palm.model,
                                           "Improve the runtime.",
                                           prompt)
cat(code.optimization)

## End(Not run)
```

Index