Package 'rtiktoken'

November 6, 2024

Title A Byte-Pair-Encoding (BPE) Tokenizer for OpenAI's Large Language Models

Version 0.0.6

Description A thin wrapper around the tiktoken-rs crate, allowing to encode text into Byte-Pair-Encoding (BPE) tokens and decode tokens back to text. This is useful to understand how Large Language Models (LLMs) perceive text.

License MIT + file LICENSE

URL https://davzim.github.io/rtiktoken/, https://github.com/DavZim/rtiktoken/

BugReports https://github.com/DavZim/rtiktoken/issues

Suggests testthat (>= 3.0.0)

SystemRequirements Cargo (Rust's package manager), rustc >= 1.65.0

Encoding UTF-8

RoxygenNote 7.3.2

Config/rextendr/version 0.3.1

Config/testthat/edition 3

Config/rtiktoken/MSRV 1.65.0

NeedsCompilation yes

Author David Zimmermann-Kollenda [aut, cre], Roger Zurawicki [aut] (tiktoken-rs Rust library), Authors of the dependent Rust crates [aut] (see AUTHORS file)

Maintainer David Zimmermann-Kollenda <david_j_zimmermann@hotmail.com>

Repository CRAN

Date/Publication 2024-11-06 15:40:02 UTC

Contents

`decode_tokens` *Decodes tokens back to text*

Description

Decodes tokens back to text

Usage

```
decode_tokens(tokens, model)
```

Arguments

`tokens`	a vector of tokens to decode, or a list of tokens
`model`	a model to use for tokenization, either a model name, e.g., `gpt-4o` or a tokenizer, e.g., `o200k_base`. See also available tokenizers.

Value

a character string of the decoded tokens or a vector or strings

See Also

`model_to_tokenizer()`, `get_tokens()`

Examples

```
tokens <- get_tokens("Hello World", "gpt-4o")
tokens
decode_tokens(tokens, "gpt-4o")

tokens <- get_tokens(c("Hello World", "Alice Bob Charlie"), "gpt-4o")
tokens
decode_tokens(tokens, "gpt-4o")
```

get_tokens *Converts text to tokens*

Description

Converts text to tokens

Usage

```
get_tokens(text, model)
```

Arguments

text	a character string to encode to tokens, can be a vector
model	a model to use for tokenization, either a model name, e.g., gpt-4o or a tokenizer, e.g., o200k_base. See also available tokenizers.

Value

a vector of tokens for the given text as integer

See Also

model_to_tokenizer(), decode_tokens()

Examples

```
get_tokens("Hello World", "gpt-4o")
get_tokens("Hello World", "o200k_base")
```

get_token_count *Returns the number of tokens in a text*

Description

Returns the number of tokens in a text

Usage

```
get_token_count(text, model)
```

Arguments

text	a character string to encode to tokens, can be a vector
model	a model to use for tokenization, either a model name, e.g., gpt-4o or a tokenizer, e.g., o200k_base. See also available tokenizers.

Value

the number of tokens in the text, vector of integers

See Also

model_to_tokenizer(), get_tokens()

Examples

```
get_token_count("Hello World", "gpt-4o")
```

---

model_to_tokenizer *Gets the name of the tokenizer used by a model*

Description

Gets the name of the tokenizer used by a model

Usage

```
model_to_tokenizer(model)
```

Arguments

model	the model to use, e.g., gpt-4o

Value

the tokenizer used by the model

Examples

```
model_to_tokenizer("gpt-4o")
model_to_tokenizer("gpt-4-1106-preview")
model_to_tokenizer("text-davinci-002")
model_to_tokenizer("text-embedding-ada-002")
model_to_tokenizer("text-embedding-3-small")
```

Index