openai / tiktoken

Type / to search

Code Issues 35 Pull requests 21 Actions Projects Security Insights

⚠ You only have a single verified email address. We recommend verifying at least one more email address to ensure you can recover your account if you lose access to your primary email.

Email settings ×

tiktoken / tiktoken_ext / openai_public.py

hauntsaninja Sync codebase ✕ 9d01e56 · 2 months ago History

```python
from tiktoken.load import data_gym_to_mergeable_bpe_ranks, load_tiktoken_bpe
```

tiktoken / tiktoken_ext / openai_public.py ↑ Top

Code Blame 127 lines (112 loc) · 4.53 KB Raw

```python
FIM_SUFFIX = "<|fim_suffix|>"
ENDOFPROMPT = "<|endofprompt|>"


def gpt2():
    mergeable_ranks = data_gym_to_mergeable_bpe_ranks(
        vocab_bpe_file="https://openaipublic.blob.core.windows.net/gpt-2/encodings/main/vocab.bpe",
        encoder_json_file="https://openaipublic.blob.core.windows.net/gpt-2/encodings/main/encoder.json",
        vocab_bpe_hash="1ce1664773c50f3e0cc8842619a93edc4624525b728b188a9e0be33b7726adc5",
        encoder_json_hash="196139668be63f3b5d6574427317ae82f612a97c5d1cdaf36ed2256dbf636783",
    )
    return {
        "name": "gpt2",
        "explicit_n_vocab": 50257,
        # The pattern in the original GPT-2 release is:
        # r"""'s|'t|'re|'ve|'m|'ll|'d| ?[\p{L}]+| ?[\p{N}]+| ?[^\s\p{L}\p{N}]+|\s+(?!\S)|\s+"""
        # This is equivalent, but executes faster:
        "pat_str": r"""'(?:[sdmt]|ll|ve|re)| ?\p{L}+| ?\p{N}+| ?[^\s\p{L}\p{N}]+|\s+(?!\S)|\s+""",
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": {ENDOFTEXT: 50256},
    }


def r50k_base():
    mergeable_ranks = load_tiktoken_bpe(
        "https://openaipublic.blob.core.windows.net/encodings/r50k_base.tiktoken",
        expected_hash="306cd27f03c1a714eca7108e03d66b7dc042abe8c258b44c199a7ed9838dd930",
    )
    return {
        "name": "r50k_base",
        "explicit_n_vocab": 50257,
        "pat_str": r"""'(?:[sdmt]|ll|ve|re)| ?\p{L}+| ?\p{N}+| ?[^\s\p{L}\p{N}]+|\s+(?!\S)|\s+""",
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": {ENDOFTEXT: 50256},
    }


def p50k_base():
    mergeable_ranks = load_tiktoken_bpe(
        "https://openaipublic.blob.core.windows.net/encodings/p50k_base.tiktoken",
        expected_hash="94b5ca7dff4d00767bc256fdd1b27e5b17361d7b8a5f968547f9f23eb70d2069",
    )
    return {
        "name": "p50k_base",
        "explicit_n_vocab": 50281,
        "pat_str": r"""'(?:[sdmt]|ll|ve|re)| ?\p{L}+| ?\p{N}+| ?[^\s\p{L}\p{N}]+|\s+(?!\S)|\s+""",
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": {ENDOFTEXT: 50256},
    }


def p50k_edit():
    mergeable_ranks = load_tiktoken_bpe(
        "https://openaipublic.blob.core.windows.net/encodings/p50k_base.tiktoken",
```

Symbols ×

Find definitions and references for functions and other symbols in this file by clicking a symbol below or in the code.

Filter symbols r

const	ENDOFTEXT
const	FIM_PREFIX
const	FIM_MIDDLE
const	FIM_SUFFIX
const	ENDOFPROMPT
func	gpt2
func	r50k_base
func	p50k_base
func	p50k_edit
func	cl100k_base
func	o200k_base
const	ENCODING_CONSTRUCT...

```python
        expected_hash="94b5ca7dff4d00767bc256fdd1b27e5b17361d7b8a5f968547f9f23eb70d2069",
    )
    special_tokens = {ENDOFTEXT: 50256, FIM_PREFIX: 50281, FIM_MIDDLE: 50282, FIM_SUFFIX: 50283}
    return {
        "name": "p50k_edit",
        "pat_str": r"""'(?:[sdmt]|ll|ve|re)| ?\p{L}+| ?\p{N}+| ?[^\s\p{L}\p{N}]+|\s+(?!\S)|\s+""",
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": special_tokens,
    }


def cl100k_base():
    mergeable_ranks = load_tiktoken_bpe(
        "https://openaipublic.blob.core.windows.net/encodings/cl100k_base.tiktoken",
        expected_hash="223921b76ee99bde995b7ff738513eef100fb51d18c93597a113bcffe865b2a7",
    )
    special_tokens = {
        ENDOFTEXT: 100257,
        FIM_PREFIX: 100258,
        FIM_MIDDLE: 100259,
        FIM_SUFFIX: 100260,
        ENDOFPROMPT: 100276,
    }
    return {
        "name": "cl100k_base",
        "pat_str": r"""'(?i:[sdmt]|ll|ve|re)|[^\r\n\p{L}\p{N}]?+\p{L}+|\p{N}{1,3}| ?[^\s\p{L}\p{N}]++[\r\n]*|\s*[\r\n]|\s+(?!\S)|\s+""",
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": special_tokens,
    }


def o200k_base():
    mergeable_ranks = load_tiktoken_bpe(
        "https://openaipublic.blob.core.windows.net/encodings/o200k_base.tiktoken",
        expected_hash="446a9538cb6c348e3516120d7c08b09f57c36495e2acfffe59a5bf8b0cfb1a2d",
    )
    special_tokens = {
        ENDOFTEXT: 199999,
        ENDOFPROMPT: 200018,
    }
    # This regex could be made more efficient
    pat_str = "|".join(
        [
            r"""[^\r\n\p{L}\p{N}]?[\p{Lu}\p{Lt}\p{Lm}\p{Lo}\p{M}]*[\p{Ll}\p{Lm}\p{Lo}\p{M}]+(?i:'s|'t|'re|'ve|'m|'ll|'d)?""",
            r"""[^\r\n\p{L}\p{N}]?[\p{Lu}\p{Lt}\p{Lm}\p{Lo}\p{M}]+[\p{Ll}\p{Lm}\p{Lo}\p{M}]*(?i:'s|'t|'re|'ve|'m|'ll|'d)?""",
            r"""\p{N}{1,3}""",
            r""" ?[^\s\p{L}\p{N}]+[\r\n/]*""",
            r"""\s*[\r\n]+""",
            r"""\s+(?!\S)""",
            r"""\s+""",
        ]
    )
    return {
        "name": "o200k_base",
        "pat_str": pat_str,
        "mergeable_ranks": mergeable_ranks,
        "special_tokens": special_tokens,
    }


ENCODING_CONSTRUCTORS = {
    "gpt2": gpt2,
    "r50k_base": r50k_base,
    "p50k_base": p50k_base,
    "p50k_edit": p50k_edit,
    "cl100k_base": cl100k_base,
    "o200k_base": o200k_base,
}
```