

# Tokenization

“tokenization”

Qwen-7B UTF-8 BPE tokenization tiktoken Qwen-7B token BPE bytes token str token

```
from transformers import AutoTokenizer

tokenizer = AutoTokenizer.from_pretrained('Qwen/Qwen-7B', trust_remote_code=True)
```

## token

token BPE UTF-8 tokenize token errors replace token UTF-8 “ ”( ) errors ignore tokenizer decode errors decode tokenizer decode errors Python .

```
>>> tokenizer.decode([51461])
'?'

>>> tokenizer.convert_ids_to_tokens([51461])
[b' \xe6\xa0']

>>> b' \xe6\xa0'.decode("utf-8", errors='replace')
'?'

>>> tokenizer.decode([51461, 117])
' '
```

```
>>> tokenizer.convert_ids_to_tokens([51461, 117])
[b' \xe6\xa0', b'\xb9']

>>> b' \xe6\xa0\xb9'.decode("utf-8", errors='replace')
'  '
```

```
bytes  token id tokenizer.get_vocab() tokenizer
token
```

## token

```
token token
tokenization token <|endoftext|>
token
token Qwen-7B <|endoftext|> Qwen-
7B-Chat <|endoftext|> <|im_start|> <|im_end|> token
<|extra_0|> <|extra_204|> str token id
tokenizer.special_tokens
```

```
(Qwen-7B Qwen-7B-Chat) bos eos unk pad mask sep
token pad token
token tokenizer token
token <|endoftext|> <|im_start|> <|im_end|> <|extra_0|> <|extra_204|>
token
```

```
from transformers import AutoTokenizer
```

```
tokenizer = AutoTokenizer.from_pretrained('Qwen/Qwen-7B', trust_remote_code=True,
pad_token='<|endoftext|>')
```

```
“ : bos eos unk
token
token
<|endoftext|> eos
token
```

```
token
```

```
token token token
token
```

```
print("<|endoftext|>")
```

tokenization

```
ids:[1350, 9639, 91, 8691, 723, 427, 91, 82598]
```

```
tokens: [b'print', b'("<', b'|', b'endo', b'ft', b'ext', b'|', b'>")]
```

```
ids: [1350, 445, 151643, 899]
```

```
tokens: [b'print', b'("', '<|endoftext|>', b'")]
```

tokenization

token

```
allowed_special=set()
```

```
>>> tokenizer('print("<|endoftext|>")', allowed_special=set())
{'input_ids': [1350, 9639, 91, 8691, 723, 427, 91, 82598], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1]}
```

allowed\_special str

```
>>> tokenizer('print("<|extra_0|>")<|endoftext|>', allowed_special={'<|endoftext|>'})
{'input_ids': [1350, 9639, 91, 15460, 62, 15, 91, 82598, 151643], 'token_type_ids': [0, 0, 0, 0, 0, 0, 0, 0, 0],
'attention_mask': [1, 1, 1, 1, 1, 1, 1, 1, 1]}
```

tokenizer


```
>>> tokenizer('print("<|extra_0|>")<|endoftext|>', allowed_special={'<|endoftext|>'},
disallowed_special=('<|extra_0|>', ))
...
ValueError: Encountered text corresponding to disallowed special token '<|extra_0|>'.
If you want this text to be encoded as a special token, pass it to `allowed_special`, e.g.
`allowed_special={'<|extra_0|>', ...}`.
If you want this text to be encoded as normal text, disable the check for this token by passing
`disallowed_special=(enc.special_tokens_set - {'<|extra_0|>'})`.
To disable this check for all special tokens, pass `disallowed_special=()`.
```

allowed\_special disallowed\_special , tiktoken

```
>>> tokenizer('print("<endoftext|>")', allowed_special="all", disallowed_special=())
{'input_ids': [1350, 445, 151643, 899], 'token_type_ids': [0, 0, 0, 0], 'attention_mask': [1, 1, 1, 1]}
```

“ [ ]  
[ ]

1.   

2. 

add\_merges.py    GitHub    qwen\_extra\_vocab.txt    token    token    qwen\_extra.tiktoken    Python

████████████████████

token████

████████████████████

WARNING - 𐀀𐀀𐀀𐀀 𐀀𐀀𐀀𐀀 would be pre-tokenized to ['𐀀𐀀𐀀 ' , ' 𐀀𐀀𐀀𐀀 '], and thus cannot be added to vocabulary

WARNING - word 𐀀𐀀 is already a token b'\xe4\xb8\x80\xe5\x8f\xaa', skipping

INFO - number of existing merges: 151643

INFO - number of words for expanding: 4

DEBUG - (b'\xe4\xb8\x80\xe5\x8f\xaa', b'\xe7\x8c\xab') (𐀀𐀀𐀀𐀀) is selected as the next merge with freq 100

DEBUG - (b'\xe5\x8f\xaa', b'\xe7\x8c\xab') (𐀀𐀀) is selected as the next merge with freq 35

DEBUG - (b'\xe6\x98\xaf\xe4\xb8\x80', b'\xe5\x8f\xaa\xe7\x8c\xab') (𐀀𐀀𐀀𐀀𐀀) is selected as the next merge with freq 35

DEBUG - (b'\xe6\x88\x91', b'\xe6\x98\xaf\xe4\xb8\x80\xe5\x8f\xaa\xe7\x8c\xab') (𐀀𐀀𐀀𐀀𐀀𐀀) is selected as the next merge with freq 20

DEBUG - (b'\xe4\xbd\xa0', b'\xe6\x98\xaf\xe4\xb8\x80\xe5\x8f\xaa\xe7\x8c\xab') (𐀀𐀀𐀀𐀀𐀀𐀀) is selected as the next merge with freq 10

DEBUG - (b'\xe4\xbb\x96', b'\xe6\x98\xaf\xe4\xb8\x80\xe5\x8f\xaa\xe7\x8c\xab') (𐀀𐀀𐀀𐀀𐀀𐀀) is selected as the next merge with freq 5

INFO - number of newly learned merges: 6

qwen\_extra.tiktoken ██████████

5LiA5Y+q54yr 151851

5Y+q54yr 151852

5piv5LiA5Y+q54yr 151853

5oiR5piv5LiA5Y+q54yr 151854

5L2g5piv5LiA5Y+q54yr 151855

5LuW5piv5LiA5Y+q54yr 151856

████████████████████

```
from transformers import AutoTokenizer
```

```
>>> tokenizer = AutoTokenizer.from_pretrained("Qwen/Qwen-7B", trust_remote_code=True,
extra_vocab_file="qwen_extra.tiktoken")
```

```
>>> len(tokenizer)
```

```
151857
```

```
>>> tokenizer("𐀀𐀀𐀀𐀀 ")
```

```
{'input_ids': [151854], 'token_type_ids': [0], 'attention_mask': [1]}
```

tokenizer 2023 10 8 extra\_vocab\_file  
qwen\_extra.tiktoken qwen.tiktoken

token

Qwen tokenizer UTF-8 tokenizer SentencePiece  
SentencePiece Unicode UTF-8

Unicode  
UTF-8

UTF-8 b'\xe4\xb8\x80\xe5\x8f\xaa' b'\x80\xe5'  
token ( b'\xe4\xb8\x80' ) ( b'\xe5\x8f\xaa' )  
token /token  
token

Unicode  
Qwen tokenizer

token token  
Qwen token / token  
UTF-8

BPE Unicode

tokenize ASCII

```
>>> tokenizer.tokenize("Panda")
```

```
[b'P', b'anda']
```

```
>>> tokenizer.tokenize(" Panda")
```

```
[b' Panda']
```

```
>>> tokenizer.tokenize("Pandas")
```

```
[b'P', b'andas']
```

```
>>> tokenizer.tokenize(" Pandas")
```

[b' Pand', b'as']

■■■■■■■■

BPE■■■■■■■■■■■■■■■■■■■■

■■■■■■■■■■■■■■■■■■■■■■■■■■■■■■

Revision #1

Created 11 January 2025 09:44:04 by Colin

Updated 11 January 2025 09:44:04 by Colin