

04-02-Create your own tokenizer

[Youtube link](#)

Attached the notebook for more info



This exercise is just to create your own tokenizer to understand how it works. (Although the internal working is still blackbox, but it gives a better idea)

You can run this in your jupyter notebook or can try go/sandbox and run there.

1- Import tiktoken package.

Official documentation: <https://github.com/openai/tiktoken> (go through its readme to understand better)

```
CreatingYourOwnTokenizer.ipynb > import tiktoken
+ Code + Markdown | ▶ Run All ↺ Restart ≡ Clear All Outputs | 📄 Variables ≡ Outline ... Python 3.10.12

# Install the tiktoken library
!pip install tiktoken

# Import the library
import tiktoken

[Z] ✓ 1.3s Python

... Requirement already satisfied: tiktoken in /usr/local/lib/python3.10/dist-packages (0.7.0)
Requirement already satisfied: regex<=2022.1.18 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2024.5.15)
Requirement already satisfied: requests<=2.26.0 in /usr/local/lib/python3.10/dist-packages (from tiktoken) (2.28.1)
Requirement already satisfied: charset-normalizer<3,>=2 in /usr/local/lib/python3.10/dist-packages (from requests<=2.26.0->tiktoken) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests<=2.26.0->tiktoken) (3.7)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests<=2.26.0->tiktoken) (1.26.19)
Requirement already satisfied: certifi<=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests<=2.26.0->tiktoken) (2024.6.2)
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager. It is recommended to use a virtual env

[notice] A new release of pip is available: 24.0 -> 24.1.1
[notice] To update, run: pip install --upgrade pip
```

2- Use the below code to create the token encoding.

```
tokenizer = tiktoken.get_encoding("cl100k_base")
#this is the model used by gpt 4 for tokenization

user_input= input("Enter Text Here")

tokens=tokenizer.encode(user_input)

print(tokens)
#This will create the tokens for your entered text

[10] ✓ 8.2s Python

... [1171, 2579]
```

3- Now you can also use the same token to decode it as shown below

```
decode = tokenizer.decode(tokens)
print(decode)
✓ 0.0s
```

This is sample text . Lets see the tokens

4- In the above one, we can't see how the text was tokenized. so we will use another function (decode_tokens_byte) to understand how text was converted to token and whether it was split while tokenizing.

```
user_input=input("")
tokens = tokenizer.encode(user_input)
decode_to_bytes = tokenizer.decode_tokens_bytes(tokens)
print(tokens)
print(decode_to_bytes)

#Tokenize this sentence and showcase the requirement . You'll see space is not a token itself but appended with the words.
# Also you'll see how thw words have been broken down.
✓ 10.9s
```

[5319, 2779, 553, 420, 11914, 323, 35883, 279, 2612]
[b'T0', b'ken', b'ize', b' this', b' sentence', b' and', b' showcase', b' the', b' output']

5- Created a sample code to highlight the tokenized text and showcase the token and character count .

```
encode = tokenizer.encode(user_input)
decode = tokenizer.decode_tokens_bytes(encode)

for token in decode:
    token_list.append(token.decode())

character_count = sum(len(i) for i in token_list)
length = len(encode)
#This is basically used for coloring each token and repeat the color after 6.
for tk in token_list:
    if count == 0:
        print('\x1b[0;47;1m' + tk + '\x1b[0m', end='')
        count+=1
    elif count == 1:
        print('\x1b[0;42;1m' + tk + '\x1b[0m', end='')
        count+=1
    elif count == 2:
        print('\x1b[0;43;1m' + tk + '\x1b[0m', end='')
        count+=1
    elif count == 3:
        print('\x1b[0;44;1m' + tk + '\x1b[0m', end='')
        count+=1
    elif count == 4:
        print('\x1b[0;46;1m' + tk + '\x1b[0m', end='')
        count+=1
    elif count == 5:
        print('\x1b[0;45;1m' + tk + '\x1b[0m', end='')
        count+=1
    else:
        count=0

print("\n\n" + str(token_list) + "\n\n")
print(str(encode) + "\n\n")
print("Token Count: " + str(length))
print("Characters: " + str(character_count))
✓ 18.0s
```

[b'This', b' is', b' ,', b' sample', b' token', b' ization', b' of', b' this', b' sentence', b' as', b' required', b' to', b' understand', b' the', b' ou', b' put', b' .', b' Also', b' we', b' will', b' color', b' this', b' sentence']

[2828, 374, 264, 6285, 4837, 2865, 315, 420, 11914, 439, 2631, 311, 3619, 279, 6833, 631, 13, 7429, 584, 690, 1933, 420, 11914]

Token Count: 23
Characters: 116