

# NLP4Web

# Practice Session 7

Tokenizers

# About me

Hovhannes Tamoyan  
(@tamohannes)



[tamohannes.com](https://tamohannes.com)



[x.com/tamohannes](https://x.com/tamohannes)

# Syllabus for Practice Sessions (PS) 7 - 12

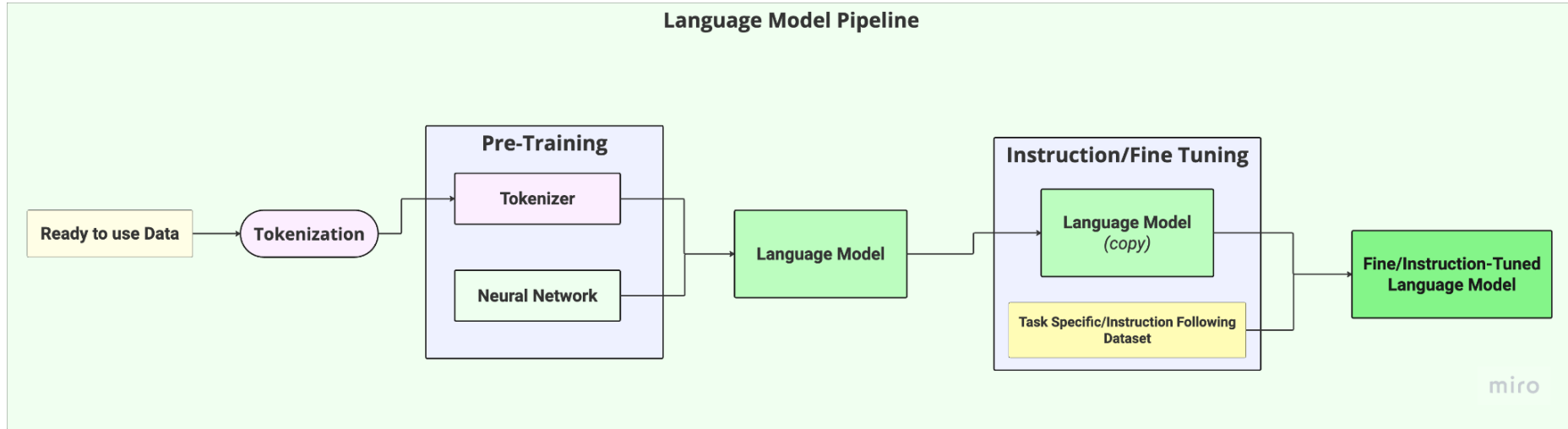
- **PS7** (05.12.2024): Tokenizers
- **PS8** (12.12.2024): Bigram, Neural Language Model and RNN
  - **HW3** release
- **PS9** (19.12.2024): Transformers Decoder-only (GPT)
- **PS10** (16.01.2025): Transformers Encoder-only (BERT) and Seq2seq Modeling
  - **HW4** release
- **PS11** (23.01.2025): Pre-Training and Fine-Tuning
- **PS12** (30.01.2025): Experiment Reproducibility and NLP Debugging
  - **HW5** release
  - **HW6** release

# The philosophy behind this series of sessions

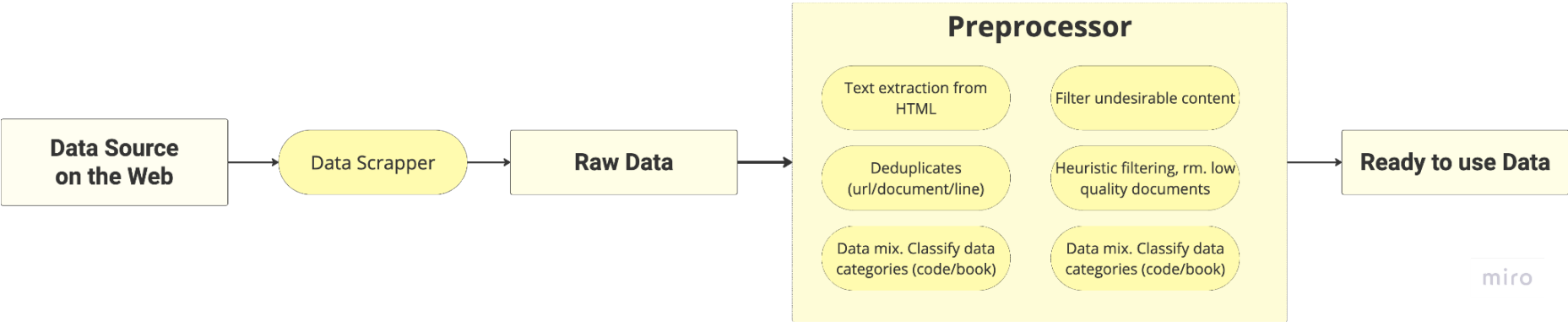
- “Build to Use”
  - Implement all the necessary components to build a language model (GPT) from scratch.
  - Use high-level wrappers for each block, as commonly practiced in industry and academia.

To not get lost in space over time, let's  
Use a **mind map**

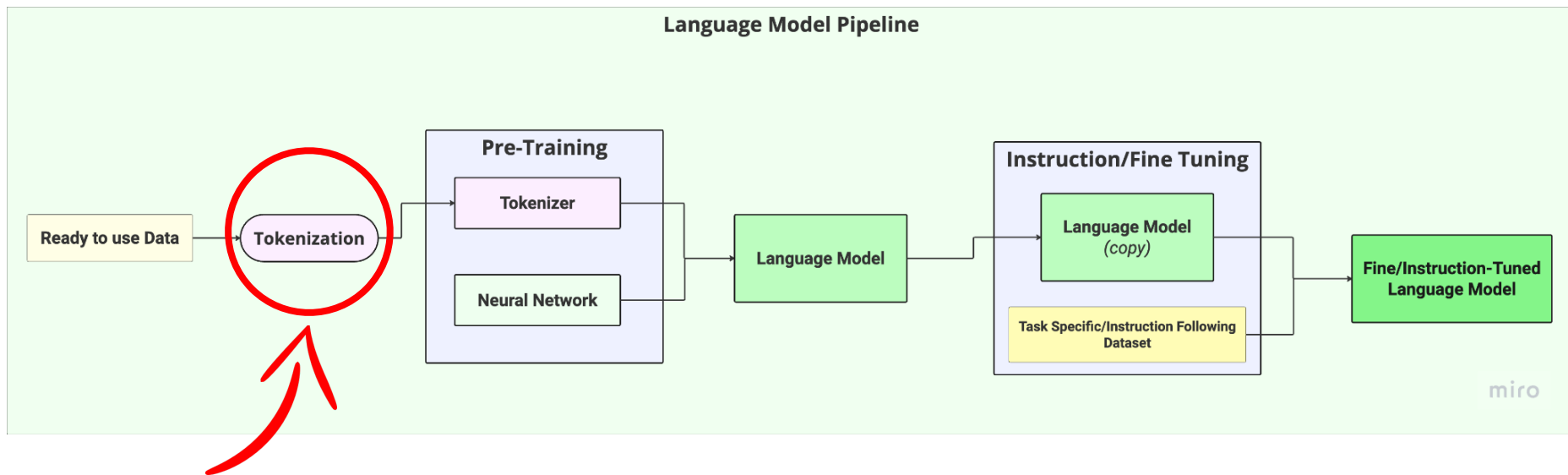
# Language Modeling pipeline nowadays



# Break down of **Ready to use Data**

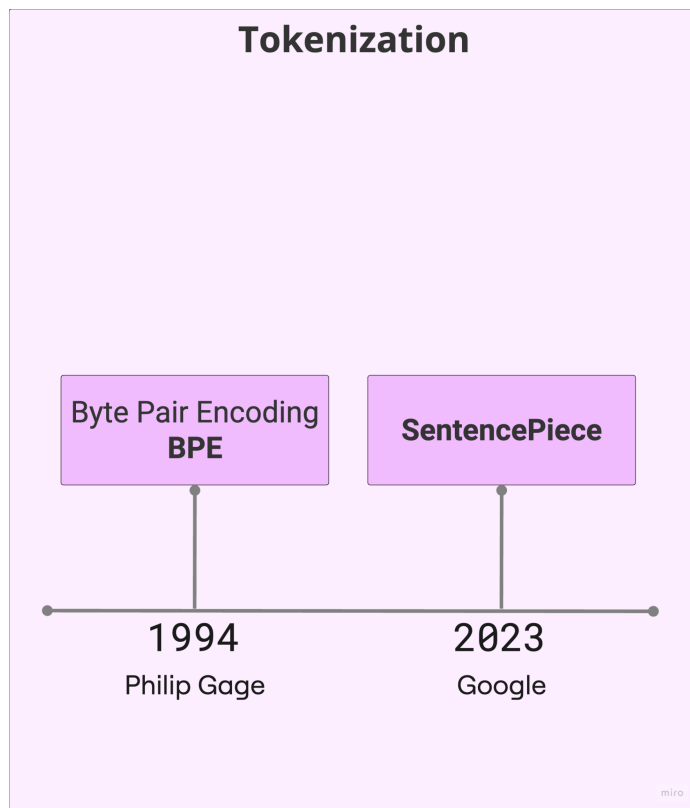


# Today's subject: **Tokenization**





# Tokenization Algorithms



Tokenization is at the heart of much of the  
weirdness in LLMs

# Why can't LLM spell words?

- The characters are divided into tokens, and some of these tokens can be quite long, such as “.DefaultCellStyle”.

# Why can't LLM spell words?

## Tiktokenizer

cl100k\_base

How many letters "l" are in the word ".DefaultCellStyle"

Token count

13

How many letters "l" are in the word ".DefaultCellStyle"

4438, 1690, 12197, 330, 75, 1, 527, 304, 279, 349  
2, 6058, 86662, 1

Show whitespace

# Why can't LLM spell words?

ChatGPT 4 

 Share

How many letters "l" are in the word ".DefaultCellStyle"



The word ".DefaultCellStyle" contains two letters "l".

# Why can't LLM do super simple string processing tasks like reversing a string?

ChatGPT 4 ▾

🔗 Share

Reverse the string ".DefaultCellStyle". Print it as a comma-separated list of characters.



To reverse the string ".DefaultCellStyle" and print it as a comma-separated list of characters, you can perform the following steps



No toolset! Just do it.



Sure! Here's the reversed string ".DefaultCellStyle" as a comma-separated list of characters:

"e, l, y, t, S, t, l, u, a, F, t, l, u, a, f, e, D, ."

Try out a live tokenizer here



[https://tiktokenizer.vercel.app/?model=o200k\\_base](https://tiktokenizer.vercel.app/?model=o200k_base)

# Why is LLM worse at non-English languages?

## Tiktokenizer

cl100k\_base

Hello how are you?  
Բարև ինչ ես

Token count  
27

Hello how are you?  
Բարև ինչ ես

9906, 1268, 527, 499, 5380, 144, 110, 145, 94, 14  
6, 222, 146, 229, 220, 145, 116, 145, 252, 145, 11  
4, 146, 223, 220, 145, 98, 145, 121

Show whitespace



# Why is LLM bad at simple arithmetic?

## Tiktokenizer

cl100k\_base

```
Multiply 12345678 by 9876543210
```

Token count  
11

```
Multiply 12345678 by 9876543210
```

```
96255, 220, 4513, 10961, 2495, 555, 220, 22207, 21  
969, 14423, 15
```

Show whitespace

Future reading: <https://www.bereni.io/2023-02-04-Integer-tokenization-is-insane/>

# Why did GPT-2 have more than necessary trouble coding in Python?

## Tiktokenizer

```
for i in range (1, 101):  
    if i % 3 == 0 and i % 5 == 0:  
        print ("FizzBuzz")  
    elif i % 3 = 0:  
        print ("Fizz")  
    elif i % 5 == 0:  
        print ("Buzz")  
    else:  
        print (i)
```

gpt2

Token count  
107

```
for i in range (1, 101):  
    if i % 3 == 0 and i % 5 == 0:  
        print ("FizzBuzz")  
    elif i % 3 = 0:  
        print ("Fizz")  
    elif i % 5 == 0:  
        print ("Buzz")  
    else:  
        print (i)
```

```
1640, 1312, 287, 2837, 357, 16, 11, 8949, 2599, 19  
8, 220, 220, 220, 611, 1312, 4064, 513, 6624, 657,  
290, 1312, 4064, 642, 6624, 657, 25, 198, 220, 22  
0, 220, 220, 220, 220, 220, 3601, 5855, 37, 6457,  
48230, 4943, 198, 220, 220, 220, 1288, 361, 1312,  
4064, 513, 796, 657, 25, 198, 220, 220, 220, 220,  
220, 220, 220, 3601, 5855, 37, 6457, 4943, 198, 22  
0, 220, 220, 1288, 361, 1312, 4064, 642, 6624, 65  
7, 25, 198, 220, 220, 220, 220, 220, 220, 220, 220,  
220, 220, 220, 220, 220, 220, 220, 220, 220, 220,  
1, 5855, 48230, 4943, 198, 220, 220, 220, 2073, 2  
5, 198, 220, 220, 220, 220, 220, 220, 220, 220, 3601, 3  
57, 72, 8
```

Show whitespace

# Why should I prefer to use YAML over JSON with LLMs?

## Tiktokenizer

```
{
  "product": {
    "type": "T-Shirt",
    "price": 20.00,
    "sizes": ["S", "M", "L"],
    "reviews": [
      { "username": "user1", "rating": 4,
        "created_at": "2023-04-19T12:30:00Z" },
      { "username": "user2", "rating": 5,
        "created_at": "2023-05-02T15:00:00Z" }
    ]
  }
}
```

Token count  
121

```
{
  "product": {
    "type": "T-Shirt",
    "price": 20.00,
    "sizes": ["S", "M", "L"],
    "reviews": [
      { "username": "user1", "rating": 4, "c
reated_at": "2023-04-19T12:30:00Z" },
      { "username": "user2", "rating": 5, "c
reated_at": "2023-05-02T15:00:00Z" }
    ]
  }
}
```

```
517, 197, 1, 3107, 794, 341, 197, 197, 45570, 794,
330, 51, 76954, 761, 197, 197, 1, 6692, 794, 220,
508, 13, 410, 345, 197, 197, 41887, 4861, 794, 448
2, 50, 498, 330, 39091, 498, 330, 43, 8257, 197, 1
97, 1, 40575, 794, 2330, 298, 197, 90, 330, 5223,
794, 330, 882, 16, 498, 330, 22696, 794, 220, 19,
11, 330, 7266, 3837, 794, 330, 2366, 18, 12, 2371,
12, 777, 51, 717, 25, 966, 25, 6726, 1, 1173, 298,
197, 90, 330, 5223, 794, 330, 882, 17, 498, 330, 2
2696, 794, 220, 20, 11, 330, 7266, 3837, 794, 330,
2366, 18, 12, 2304, 12, 2437, 51, 868, 25, 410, 2
5, 410, 57, 1, 457, 197, 197, 933, 197, 534, 92
```

Show whitespace

## Tiktokenizer

```
product:
  type: T-Shirt
  price: 20.00
  sizes:
    - S
    - M
    - L
  reviews:
    - username: user1
      rating: 4
      created_at: "2023-04-19T12:30:00Z"
    - username: user2
      rating: 5
      created_at: "2023-05-02T15:00:00Z"
```

Token count  
104

```
product:
  type: T-Shirt
  price: 20.00
  sizes:
    - S
    - M
    - L
  reviews:
    - username: user1
      rating: 4
      created_at: "2023-04-19T12:30:00Z"
    - username: user2
      rating: 5
      created_at: "2023-05-02T15:00:00Z"
```

```
3107, 512, 13459, 25, 350, 76954, 198, 88219, 25,
220, 508, 13, 410, 198, 1942, 4861, 512, 197, 197,
12, 328, 198, 197, 197, 12, 386, 198, 197, 197, 82
88, 198, 17643, 5182, 512, 197, 197, 12, 6059, 25,
1217, 16, 198, 298, 7145, 1113, 25, 220, 19, 198,
298, 197, 7266, 3837, 25, 330, 2366, 18, 12, 2371,
12, 777, 51, 717, 25, 966, 25, 410, 57, 702, 197,
197, 12, 6059, 25, 1217, 17, 198, 298, 7145, 1113,
25, 220, 20, 198, 298, 197, 7266, 3837, 25, 330, 2
366, 18, 12, 2304, 12, 2437, 51, 868, 25, 410, 25,
410, 57, 1
```

Show whitespace

LLM is not end-to-end language modeling

What is the real root of suffering?

What is the real root of suffering?

Tokenization

Let's jump into it 🚀