MACHADO BOT: GENERATING TEXTS LIKE MACHADO DE ASSIS

An Introduction to Language Models

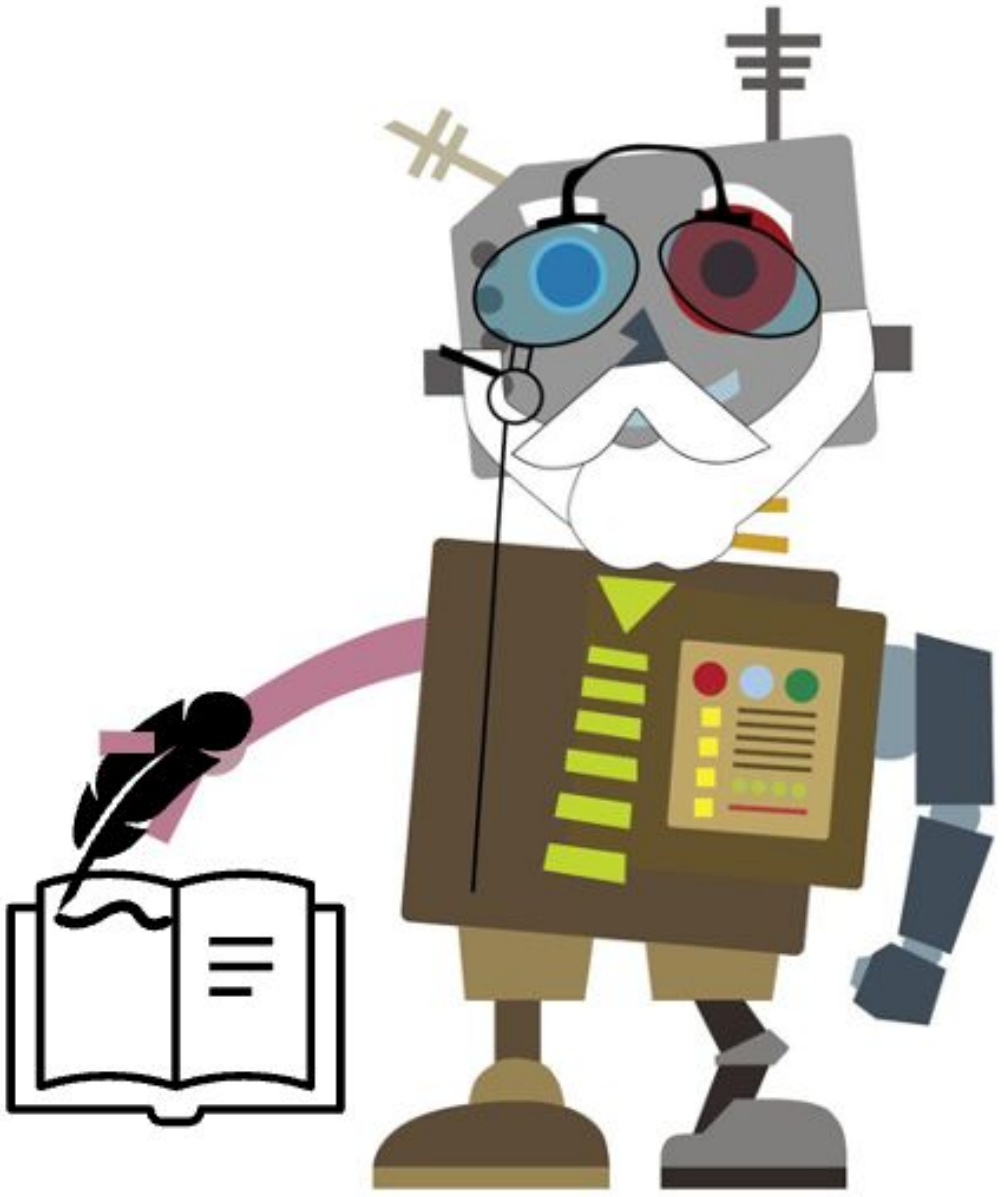
Prof. Marcelo Rovai, UNIFEI

MACHADO BOT

What is Machado Bot?

A model trained to generate text in the style of Machado de Assis. He uses texts extracted from books such as *Dom Casmurro* and *Memórias Póstumas de Braz Cubas*.

Simplified introduction to Large Language Models (LLMs) such as GPT.



https://github.com/Mjrovai/MachadoAssisBot

DATA PREPARATION

Data was collected from 7 books by Machado de Assis (2.4 million characters).

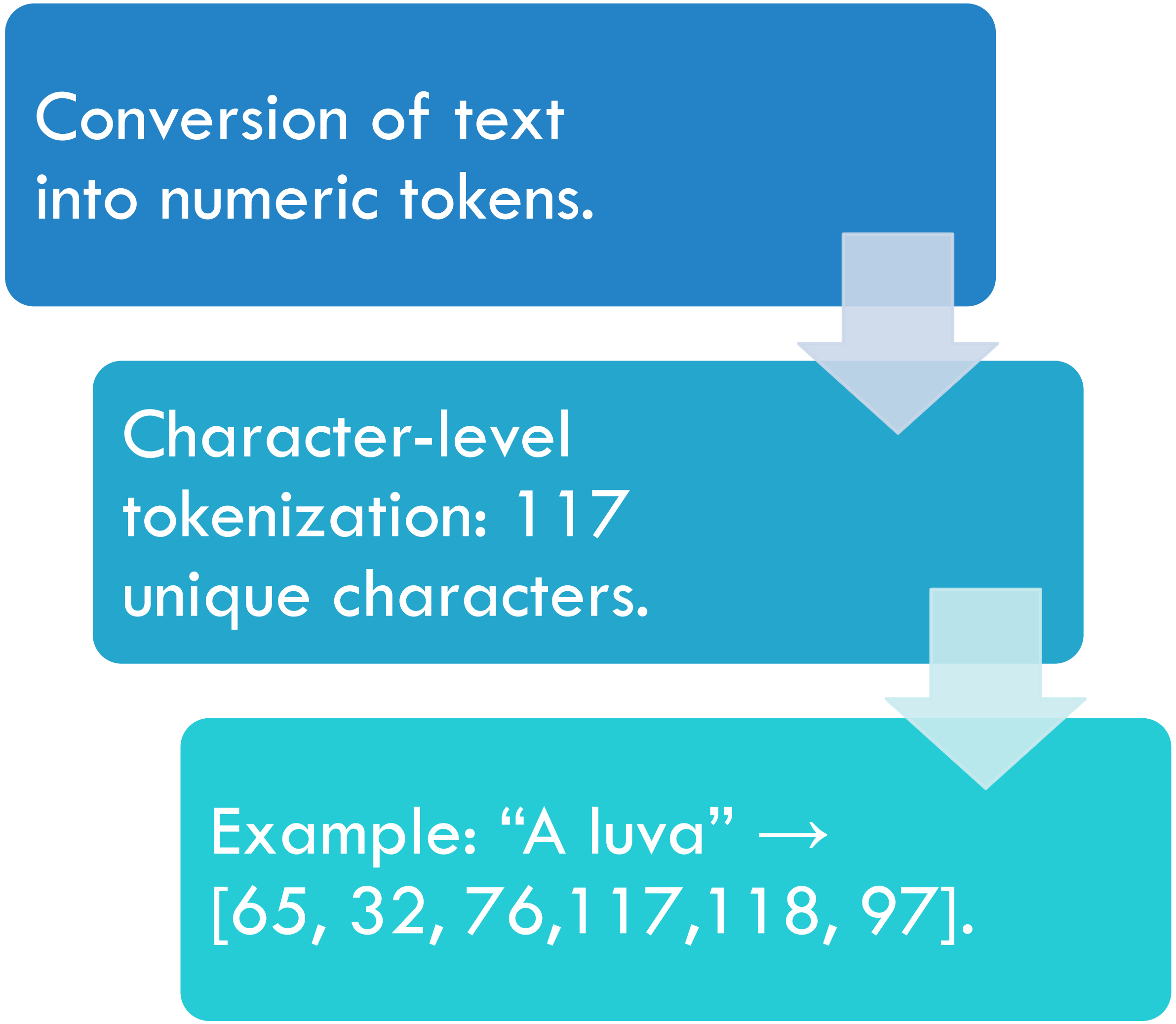
Preprocessing: Removal of irrelevant characters and structuring of the text for analysis.

Importance of clean data for training.

Project Gutenberg

Memorias Posthumas de Braz Cubas, Dom Casmurro, Quincas Borba, Papeis Avulsos, A Mão e a Luva, Esaú e Jacob, and *Memorial de Ayres*.

TOKENIZATION AND VOCABULARY

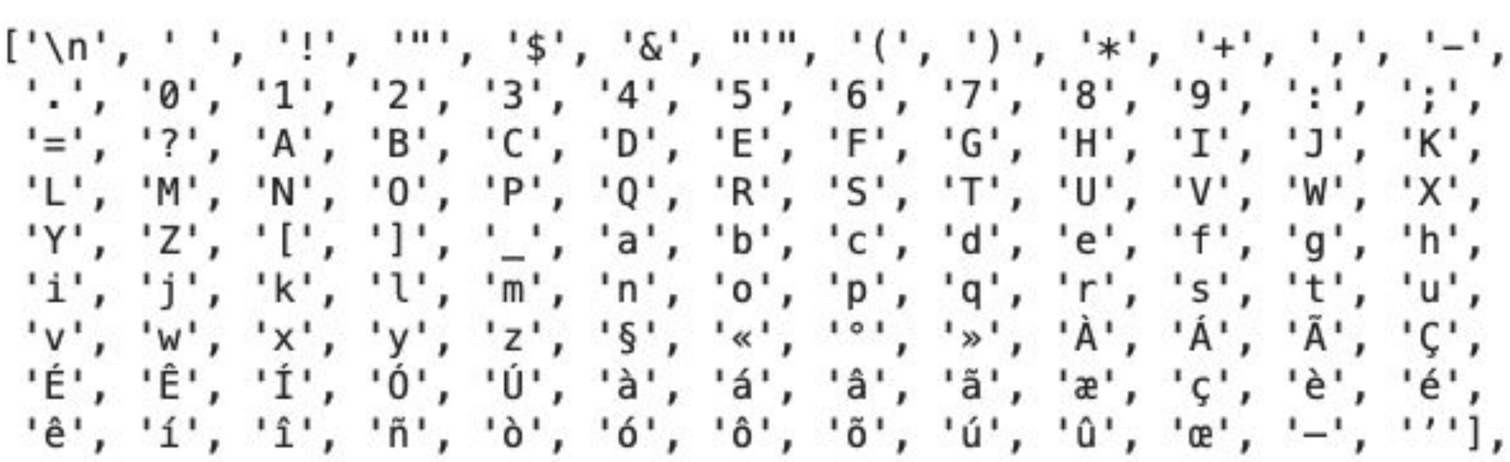


Conversion of text into numeric tokens.

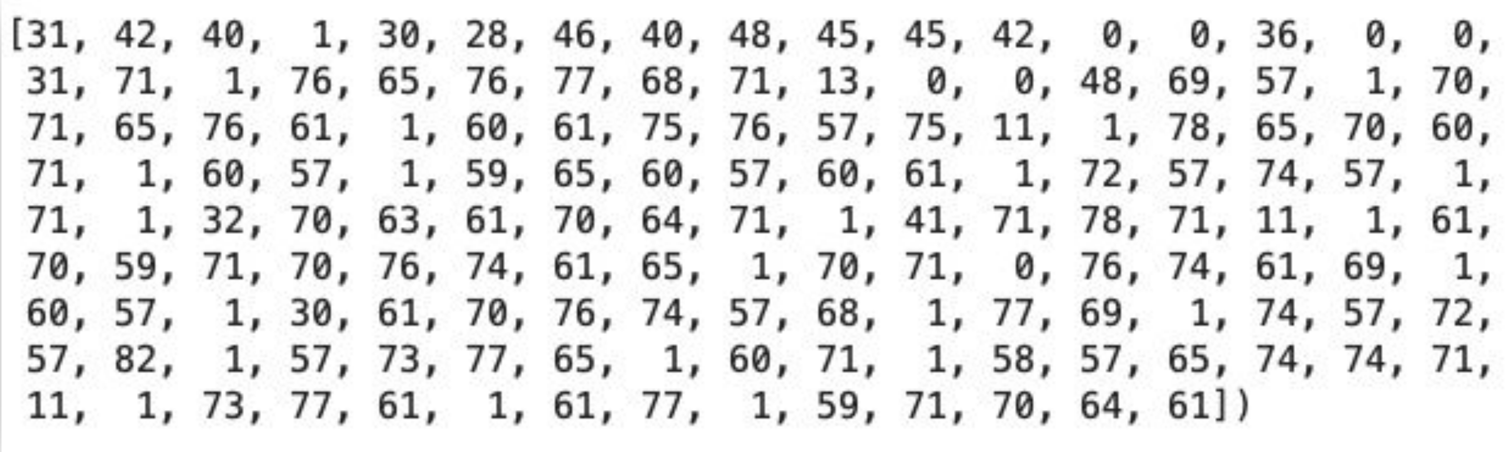
Character-level tokenization: 117 unique characters.

Example: “A luva” → [65, 32, 76,117,118, 97].

TOKENIZATION



```
['\n', ' ', '!', '"', '$', '&', "'", '(', ')', '*', '+', ',', '-',
 '.', '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', ':', ';',
 '=', '?', 'A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K',
 'L', 'M', 'N', 'O', 'P', 'Q', 'R', 'S', 'T', 'U', 'V', 'W', 'X',
 'Y', 'Z', '[', ']', '_', 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h',
 'i', 'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r', 's', 't', 'u',
 'v', 'w', 'x', 'y', 'z', '§', '«', '°', '»', 'À', 'Á', 'Ã', 'Ç',
 'É', 'Ê', 'Í', 'Ó', 'Ú', 'à', 'á', 'â', 'ã', 'æ', 'ç', 'è', 'é',
 'ê', 'í', 'î', 'ñ', 'ò', 'ó', 'ô', 'õ', 'ú', 'û', 'œ', '—', '’'],
```



```
[31, 42, 40,  1, 30, 28, 46, 40, 48, 45, 45, 42,  0,  0, 36,  0,  0,
 31, 71,  1, 76, 65, 76, 77, 68, 71, 13,  0,  0, 48, 69, 57,  1, 70,
 71, 65, 76, 61,  1, 60, 61, 75, 76, 57, 75, 11,  1, 78, 65, 70, 60,
 71,  1, 60, 57,  1, 59, 65, 60, 57, 60, 61,  1, 72, 57, 74, 57,  1,
 71,  1, 32, 70, 63, 61, 70, 64, 71,  1, 41, 71, 78, 71, 11,  1, 61,
 70, 59, 71, 70, 76, 74, 61, 65,  1, 70, 71,  0, 76, 74, 61, 69,  1,
 60, 57,  1, 30, 61, 70, 76, 74, 57, 68,  1, 77, 69,  1, 74, 57, 72,
 57, 82,  1, 57, 73, 77, 65,  1, 60, 71,  1, 58, 57, 65, 74, 74, 71,
 11,  1, 73, 77, 61,  1, 61, 77,  1, 59, 71, 70, 64, 61])
```

https://platform.openai.com/tokenizer

TRAINING SEQUENCES

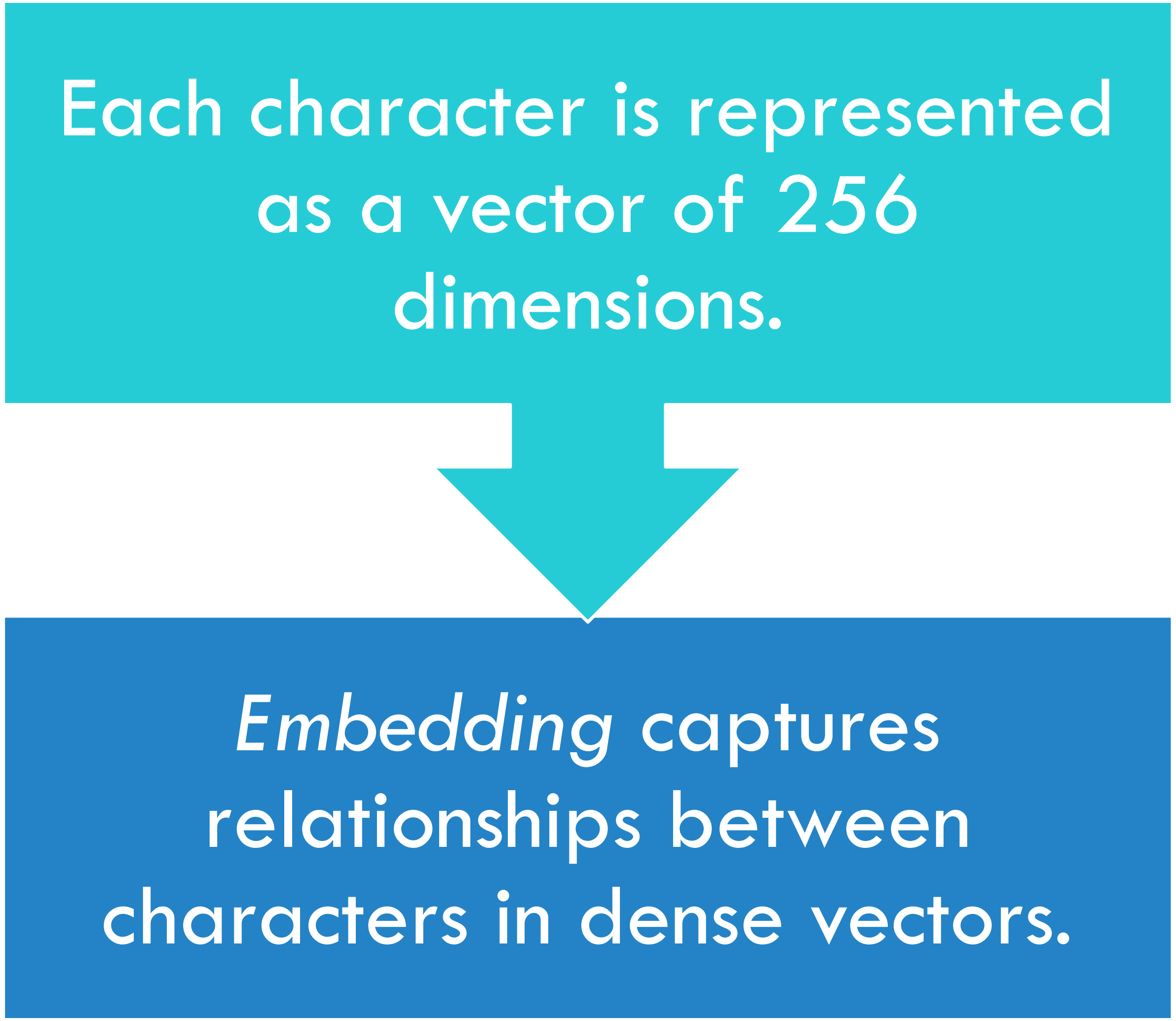
Goal: Predict the next character in a sequence.

Length of the sequence: 150 characters (paragraph).

Input 'Boa tarde, meu nom’

Output 'oa tarde, meu nome'.

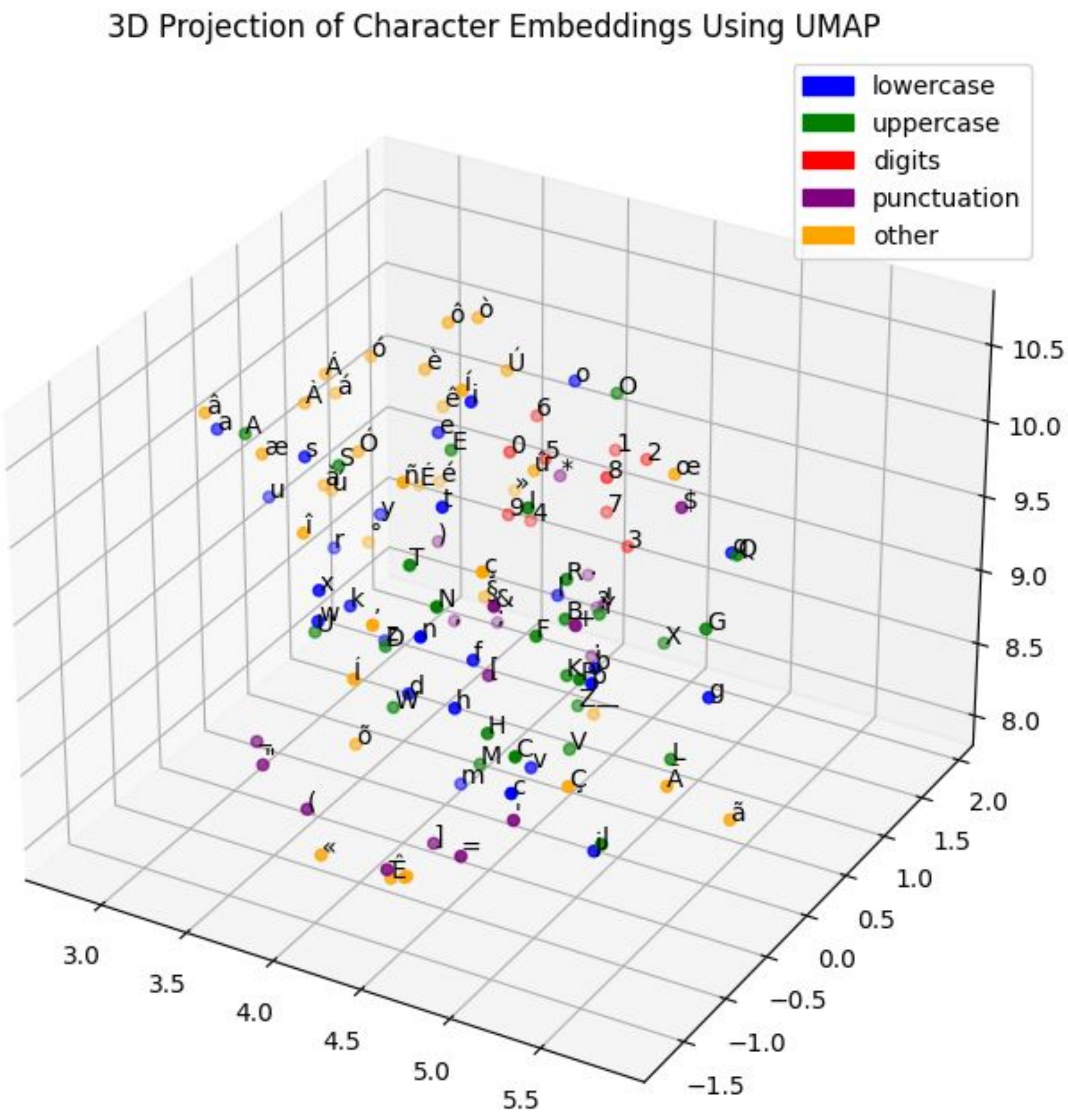
EMBEDDING



Each character is represented as a vector of 256 dimensions.

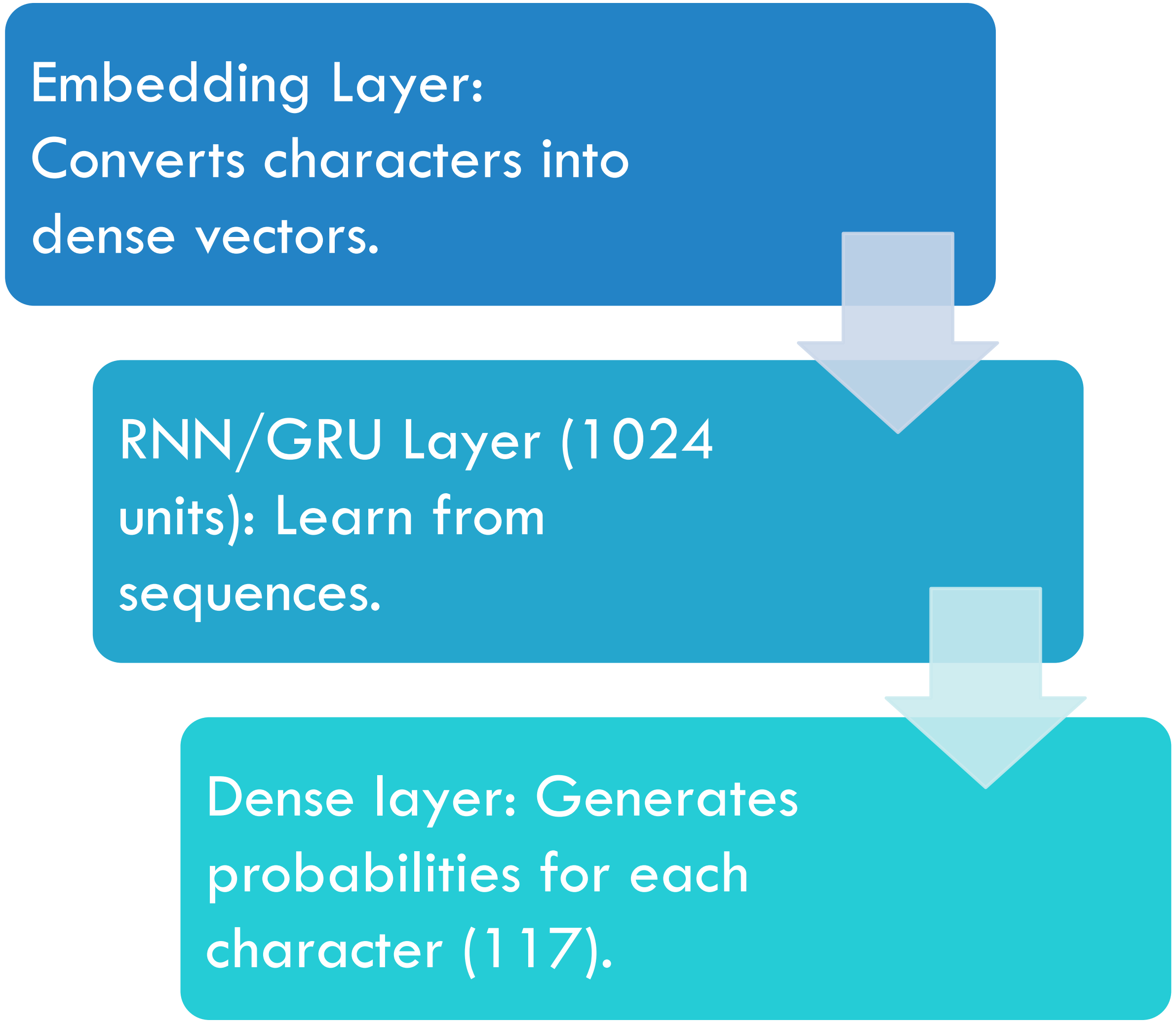
Embedding captures relationships between characters in dense vectors.

EMBEDDING



Word2Vec - Embedding Projector

MODEL ARCHITECTURE



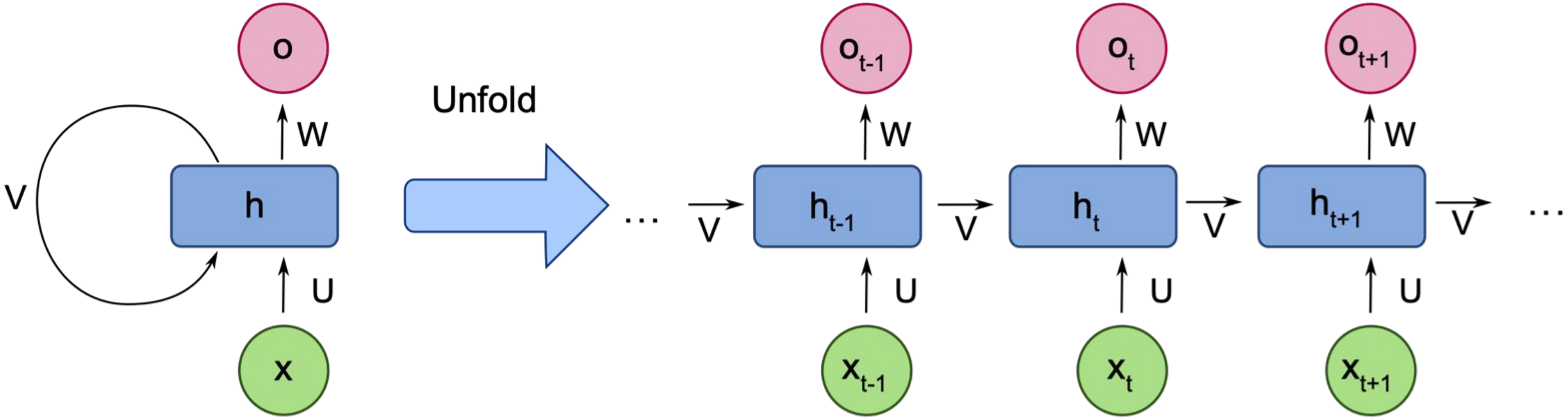
Embedding Layer: Converts characters into dense vectors.

RNN/GRU Layer (1024 units): Learn from sequences.

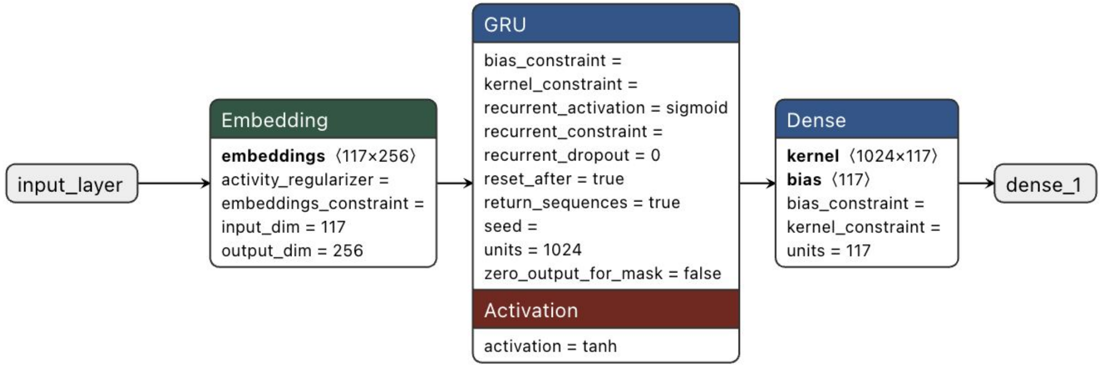
Dense layer: Generates probabilities for each character (117).

Deep Learning models (or artificial neural networks)

Recurrent Neural Networks (RNNs): Designed for **sequential data like time series or text**, these networks use their internal state (memory) to process sequences of inputs.



RNN MODEL (RECURRENT)



input_layer

Embedding

embeddings ⟨117×256⟩
activity_regularizer =
embeddings_constraint =
input_dim = 117
output_dim = 256

GRU

bias_constraint =
kernel_constraint =
recurrent_activation = sigmoid
recurrent_constraint =
recurrent_dropout = 0
reset_after = true
return_sequences = true
seed =
units = 1024
zero_output_for_mask = false

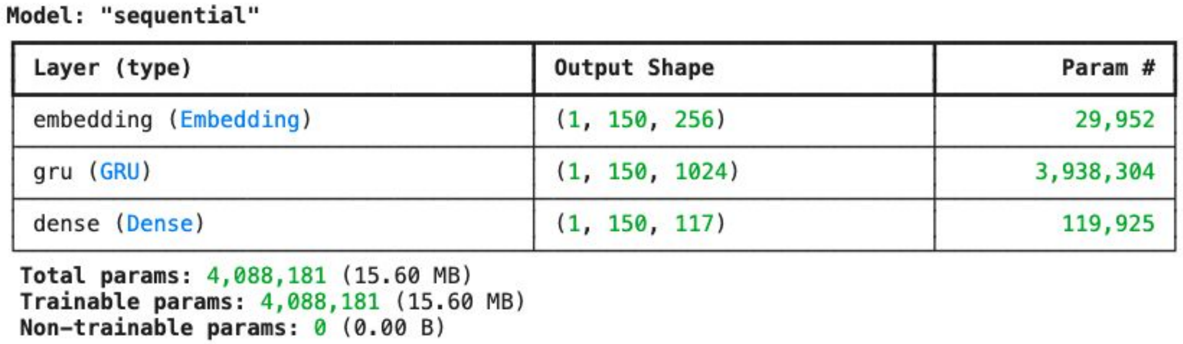
Activation

activation = tanh

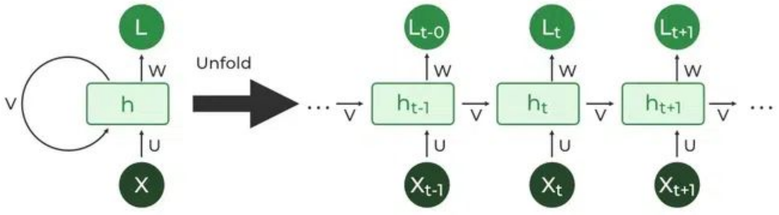
Dense

kernel ⟨1024×117⟩
bias ⟨117⟩
bias_constraint =
kernel_constraint =
units = 117

dense_1



```
Model: "sequential"
┌─────────────────────────────┬──────────────────────┬─────────────┐
│ Layer (type)                │ Output Shape         │     Param # │
├─────────────────────────────┼──────────────────────┼─────────────┤
│ embedding (Embedding)       │ (1, 150, 256)        │      29,952 │
├─────────────────────────────┼──────────────────────┼─────────────┤
│ gru (GRU)                   │ (1, 150, 1024)       │   3,938,304 │
├─────────────────────────────┼──────────────────────┼─────────────┤
│ dense (Dense)               │ (1, 150, 117)        │     119,925 │
└─────────────────────────────┴──────────────────────┴─────────────┘
 Total params: 4,088,181 (15.60 MB)
 Trainable params: 4,088,181 (15.60 MB)
 Non-trainable params: 0 (0.00 B)
```



MODEL TRAINING

Loss Function: Categorical Sparse Crossentropy

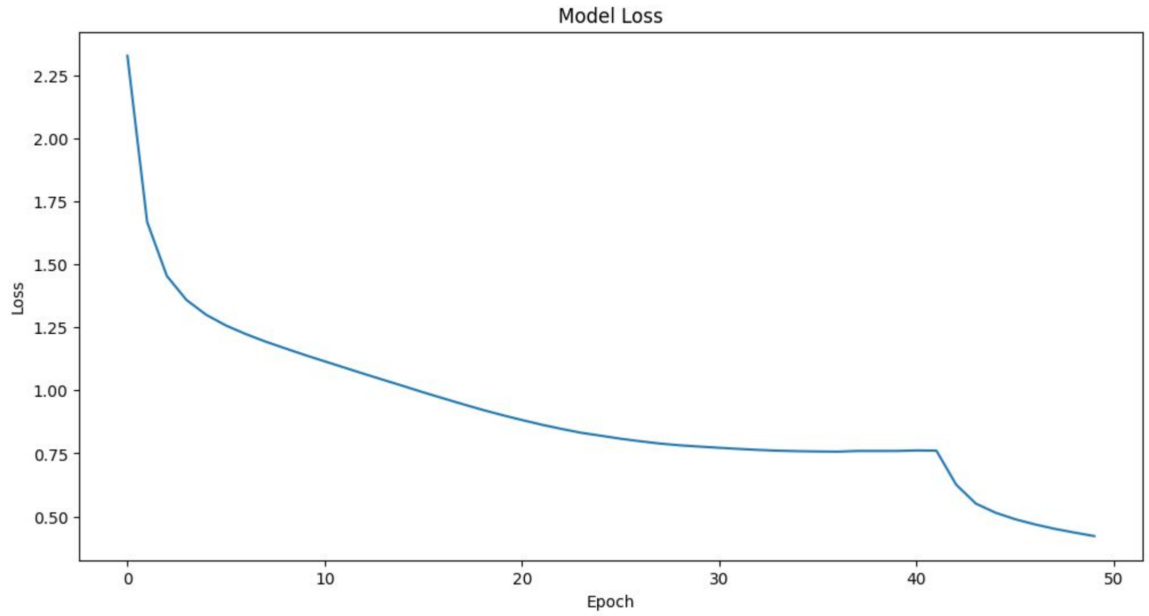
Optimizer: Adam

Epochs:50

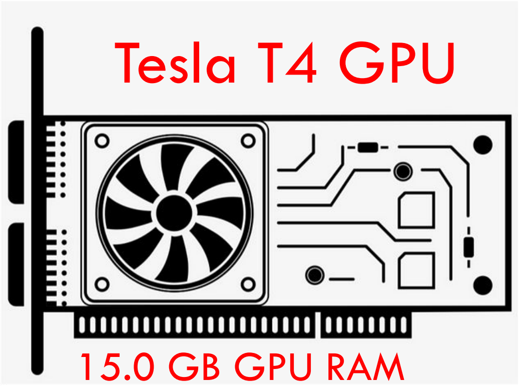
lot size: 64

buffer size: 10,000

Monitoring loss reduction over time.



(20 to 25 minutes for training)

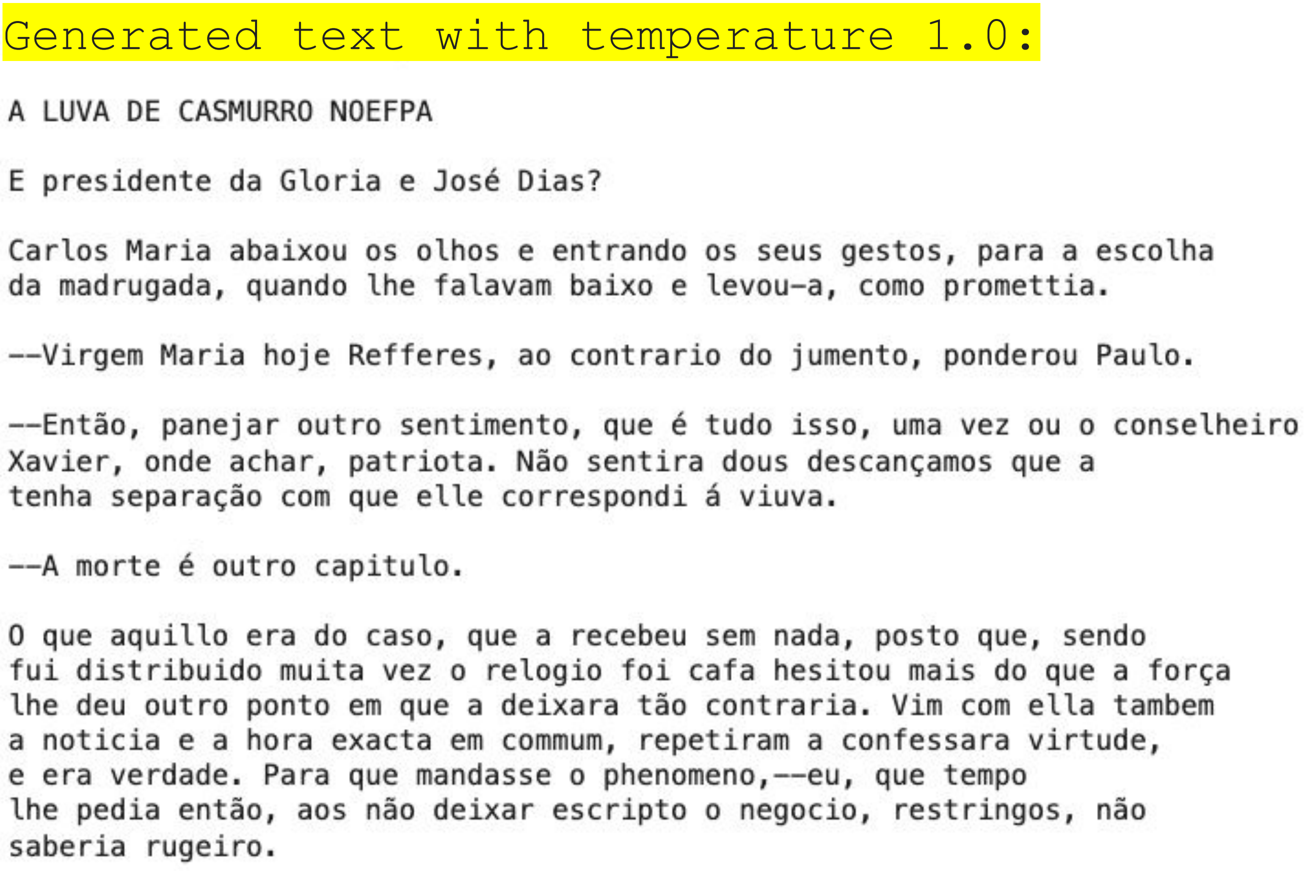


TEXT GENERATION

The template generates text character by character from an initial text:

"A LUVA DE CASMURRO".

Temperature controls randomness (0.5 for predictable, 1.0 for creative text).



```
Generated text with temperature 1.0:

A LUVA DE CASMURRO NOEFPA

E presidente da Gloria e José Dias?

Carlos Maria abaixou os olhos e entrando os seus gestos, para a escolha
da madrugada, quando lhe falavam baixo e levou-a, como promettia.

--Virgem Maria hoje Refferes, ao contrario do jumento, ponderou Paulo.

--Então, panejar outro sentimento, que é tudo isso, uma vez ou o conselheiro
Xavier, onde achar, patriota. Não sentira dous descançamos que a
tenha separação com que elle correspondi á viuva.

--A morte é outro capitulo.

O que aquillo era do caso, que a recebeu sem nada, posto que, sendo
fui distribuido muita vez o relogio foi cafa hesitou mais do que a força
lhe deu outro ponto em que a deixara tão contraria. Vim com ella tambem
a noticia e a hora exacta em commum, repetiram a confessara virtude,
e era verdade. Para que mandasse o phenomeno,--eu, que tempo
lhe pedia então, aos não deixar escripto o negocio, restringos, não
saberia rugeiro.
```

CHALLENGES AND LIMITATIONS

Limited context window (150 characters).

Difficulty in maintaining coherence in long texts.

Character-level modeling vs. word-level modeling.

CONNECTING WITH MODERN LANGUAGE MODELS

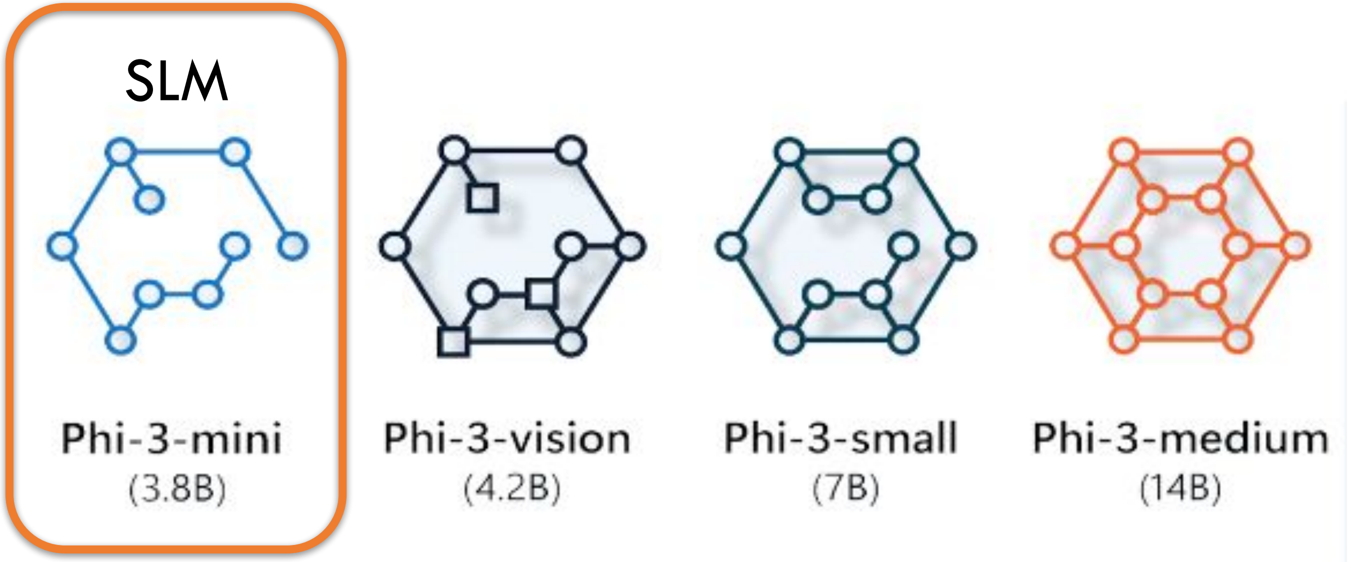
Our Model (MachadoBot) :

- **Training data: 2.4 million characters (bytes) (7 books).**
  4 million parameters,
  Character-level tokenization (150)
  RNN architecture.

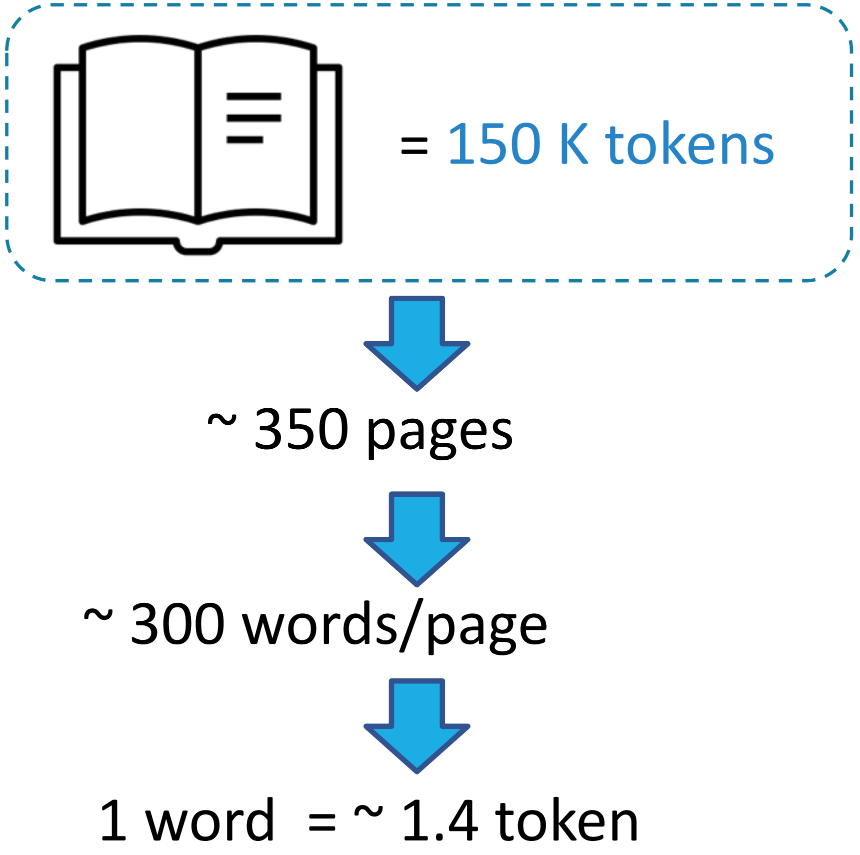
Open AI GPT-3 (2020):

- **Training data: 45 Trillion bytes (text)**
  175 billion parameters,
  Subword tokenization (2,048 tokens),
  Transformer Architecture.

Modern models handle long-range dependencies better.



- **Architecture: Transformer – 3.8 Billion Parameters**
  Inputs: Text.
  Context length: 128k tokens
  GPU: 512 H100-80G
  Training time: 7 days
  Training data: 3.3 Trillion tokens**



= 150 K tokens

~ 350 pages

~ 300 words/page

1 word = ~ 1.4 token

** Equivalent to 23 million books, that is:
17% of All the books in the world

Questions?

TINYML4D