ReaLLM ASIC

Make Your Own Lightweight LLMs

Outline of the Tutorial

- Intro
  - LLMs and Applications
  - Recap of LLM architecture
- Hands-on AI From Scratch:
  - Building and training a custom lightweight LLM
  - Data preparation and preprocessing
  - Model training options and optimization
- Checkpoints and Finetuning

LLM Capabilities and Datasets

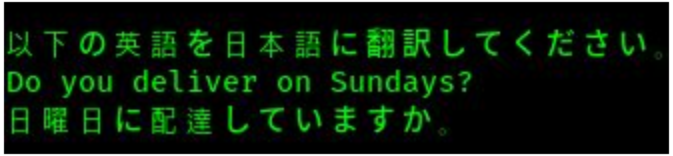
LLM Training - “Next Token Prediction”

LLMs do not require us to label the correct data, it simply trains on datasets.

To improve, it corrects itself to get better at prediction of the next element, for example:

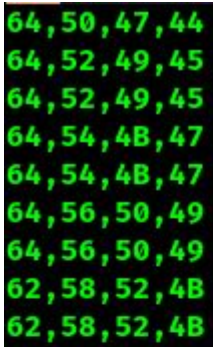
- Music
  - Prediction of the next note
- Translation
  - Prediction of next word
- Mathematics
  - Prediction of next number

Translation Dataset



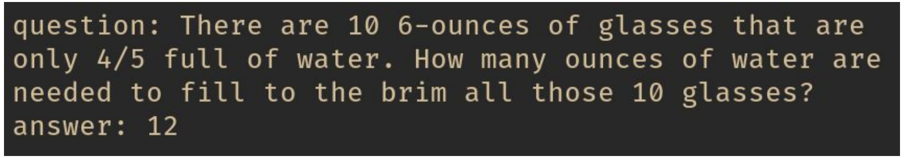
```
以下の英語を日本語に翻訳してください。
Do you deliver on Sundays?
日曜日に配達していますか。
```

Midi Dataset



```
64,50,47,44
64,52,49,45
64,52,49,45
64,54,4B,47
64,54,4B,47
64,56,50,49
64,56,50,49
62,58,52,4B
62,58,52,4B
```

Math Dataset



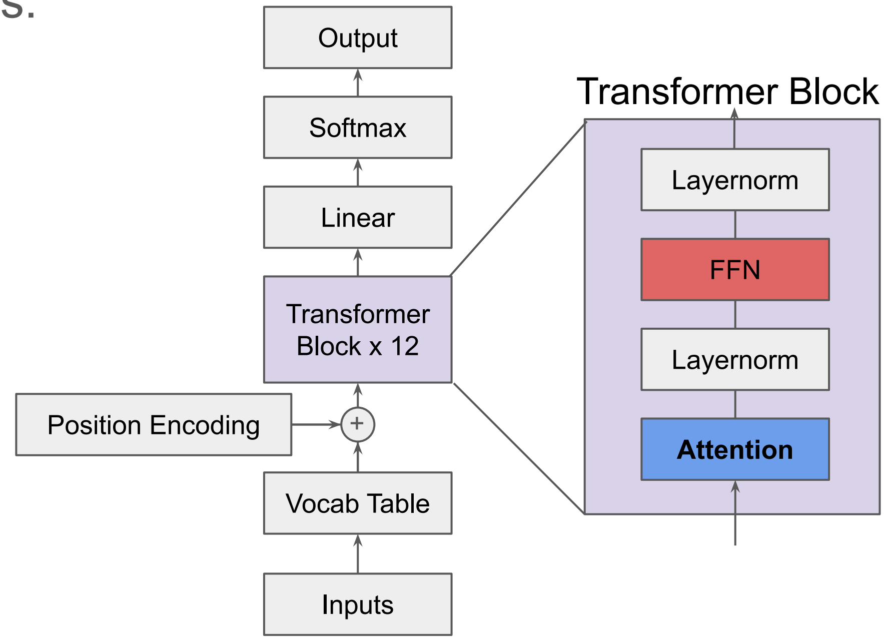
```
question: There are 10 6-ounces of glasses that are
only 4/5 full of water. How many ounces of water are
needed to fill to the brim all those 10 glasses?
answer: 12
```

LLM Applications And Datasets

One architecture, limited only by datasets:

- Translation
- Poetry
- Music
- Robotics Motion
- Cooking Recipes
- Editing and Writing Reports

LLM Transformer Architecture



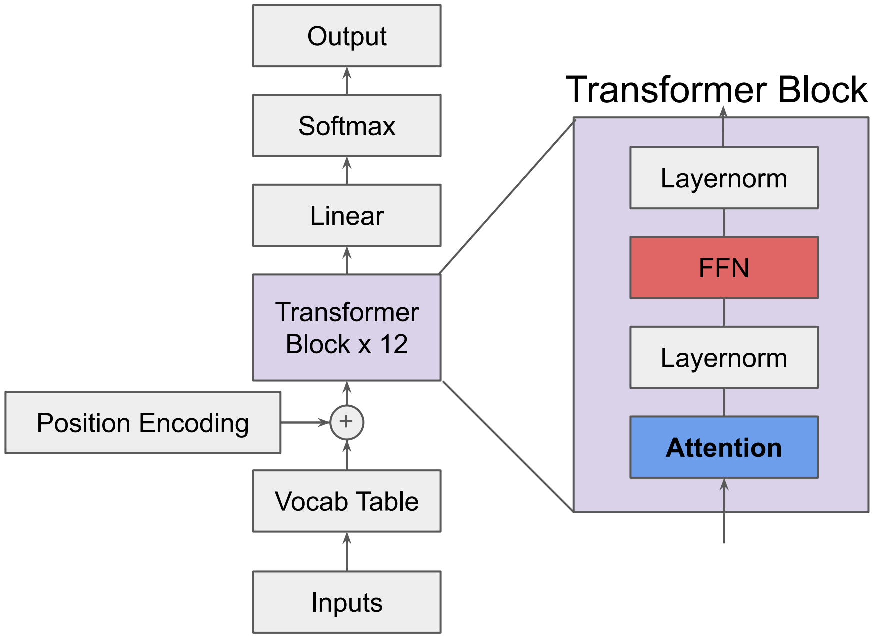
Settings When Training LLMs

Major Hyperparameters

- "Height" - Number of Layers
  - Deep Networks -> Abstract Knowledge
  - Linearly increases size of network

- "Width" - Dimensions per Token
  - Better Per Token Understanding
  - Non-linear increase in size

LLM Transformer Architecture



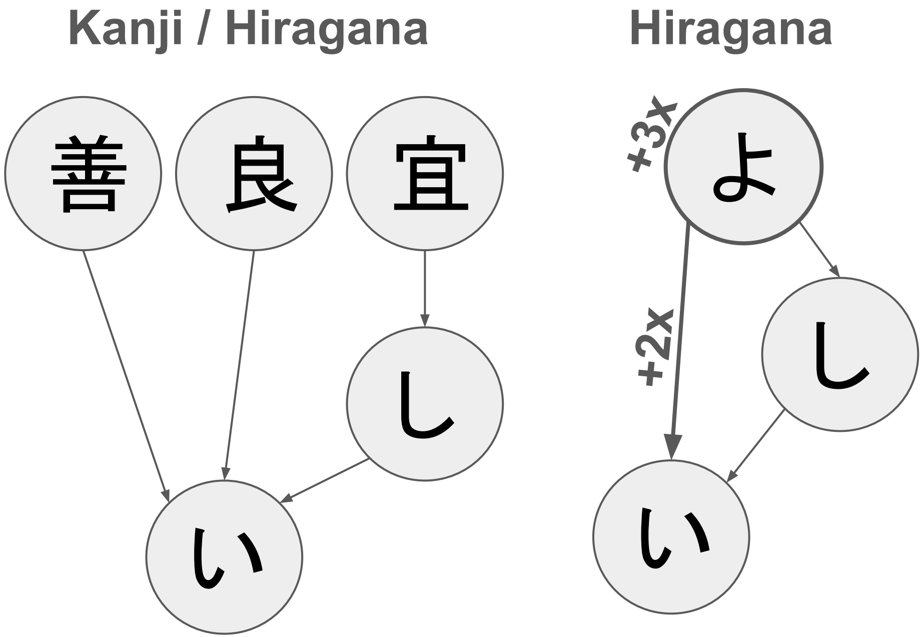
Tokenization Settings

Tokens need to gain “Experience Points”:

- Higher frequency of a token in the dataset then the easier it is an LLM to learn about it.

Language Example:

- Preprocessing Kanji -> Hiragana:
    - Each Hiragana is a Token
    - Faster training (smaller model)
    - Faster accumulation of “experience points”

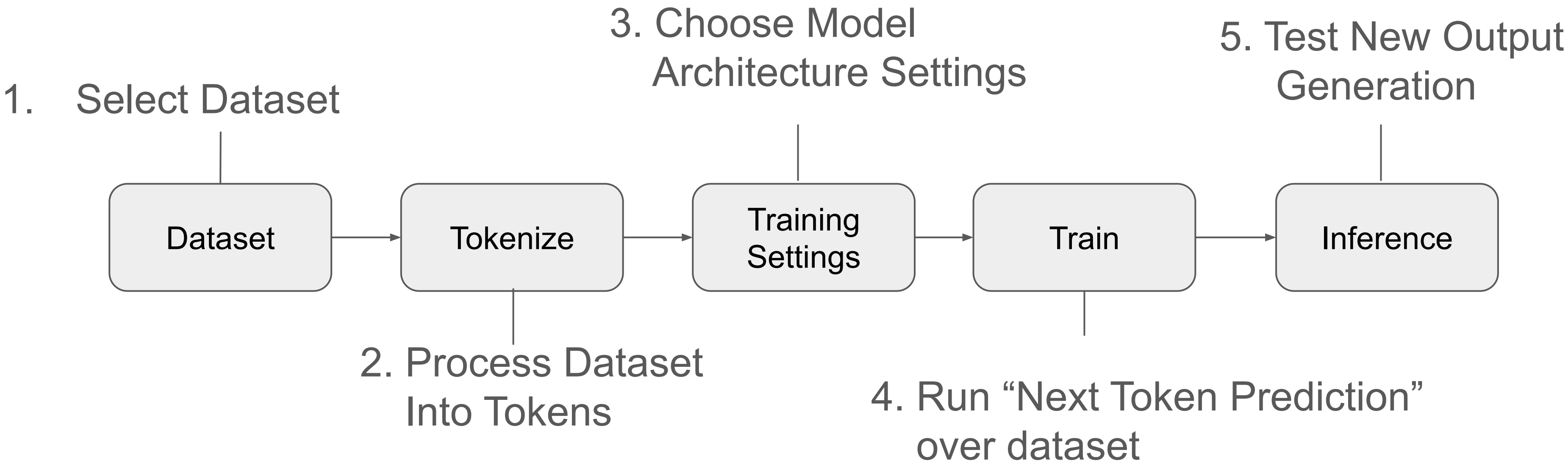


Datasets

We are limited only by the datasets we can train on.

- Music
  - We’ll work with JS Bach
  - Tokenization: CSV with Base 12 to represent Octaves
- Language
  - Japanese <-> English Translation Pairs
  - Tokenization: Hiragana, Katakana, English and Punctuation.
- Generation
  - Shakespeare
  - Tokenization: Letters

Standard LLM Training Recipe



Dataset → Tokenize → Training Settings → Train → Inference

1. Select Dataset

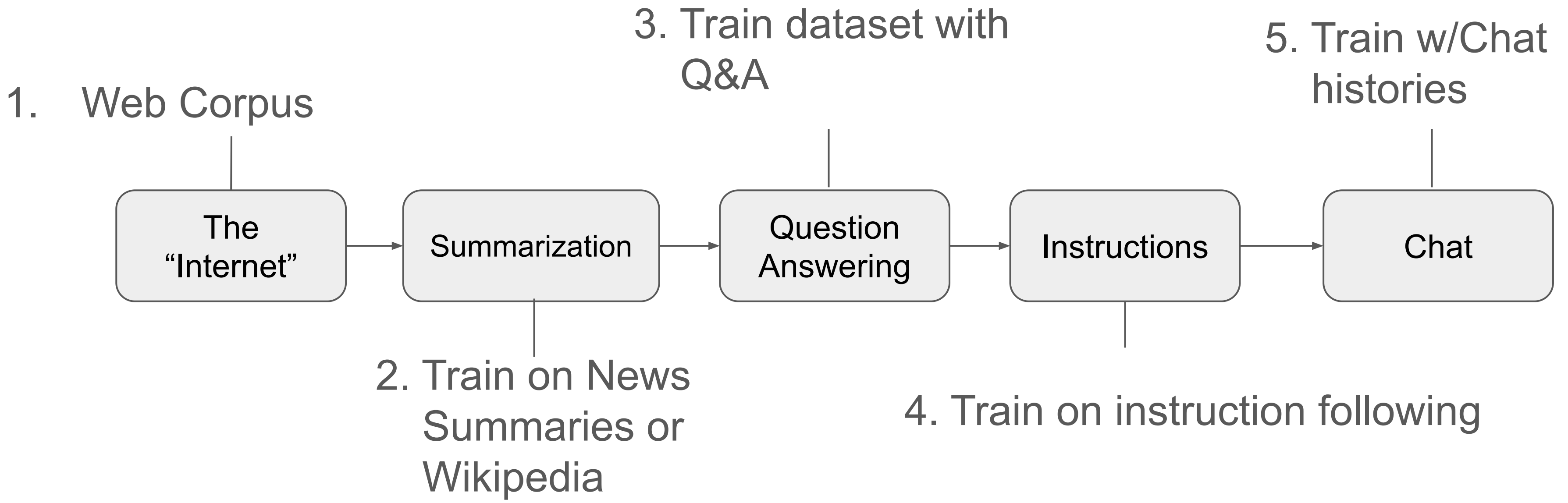
2. Process Dataset Into Tokens

3. Choose Model Architecture Settings

4. Run “Next Token Prediction” over dataset

5. Test New Output Generation

Curriculums - Dataset Training Order



1. Web Corpus

2. Train on News Summaries or Wikipedia

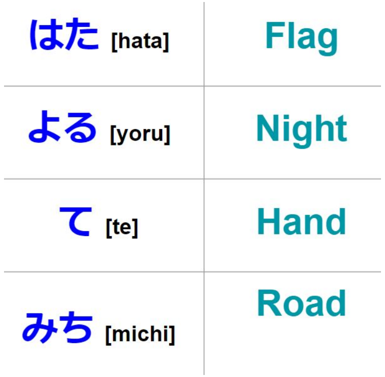
3. Train dataset with Q&A

4. Train on instruction following

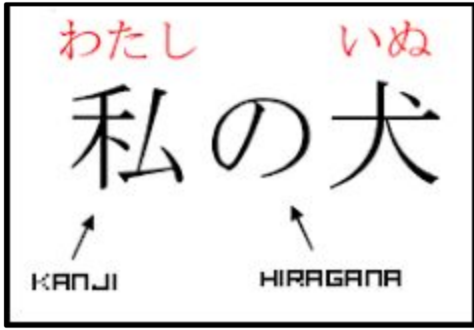
5. Train w/Chat histories

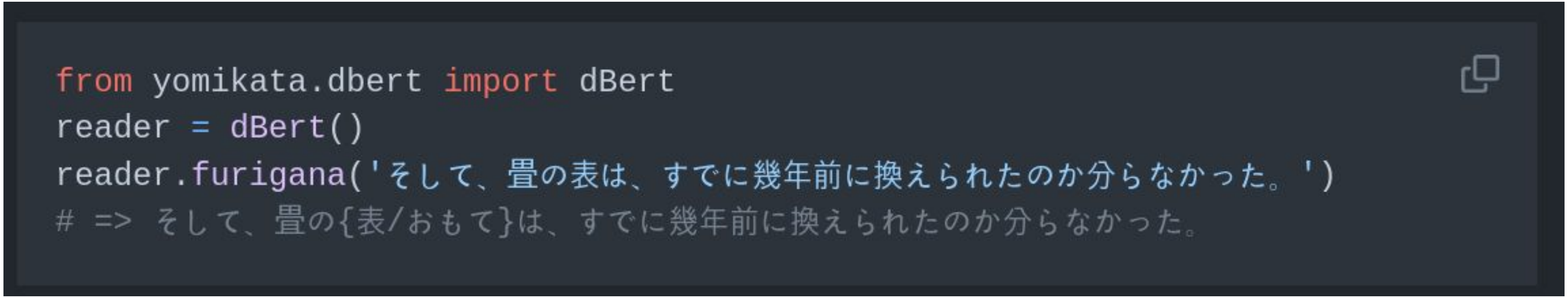
Kanji to Hiragana Preprocessors

- ***Hiragana is essentially phonetic***
- Kanji -> Hiragana Processors Exist
  - https://github.com/passaglia/yomikata
  - https://github.com/morikatron/yakinori
- *-> Phonetic sub-char approach*



はた [hata]	Flag
よる [yoru]	Night
て [te]	Hand
みち [michi]	Road

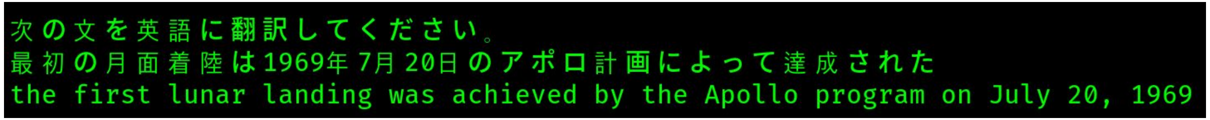




```
from yomikata.dbert import dBert
reader = dBert()
reader.furigana('そして、畳の表は、すでに幾年前に換えられたのか分らなかった。')
# => そして、畳の{表/おもて}は、すでに幾年前に換えられたのか分らなかった。
```

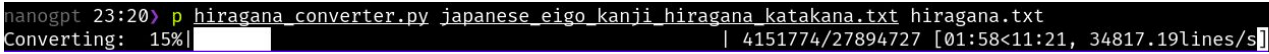
Pre-process Into Individual Numbers, Letters, Hiragana, Katakana

- Language Translation Pairs



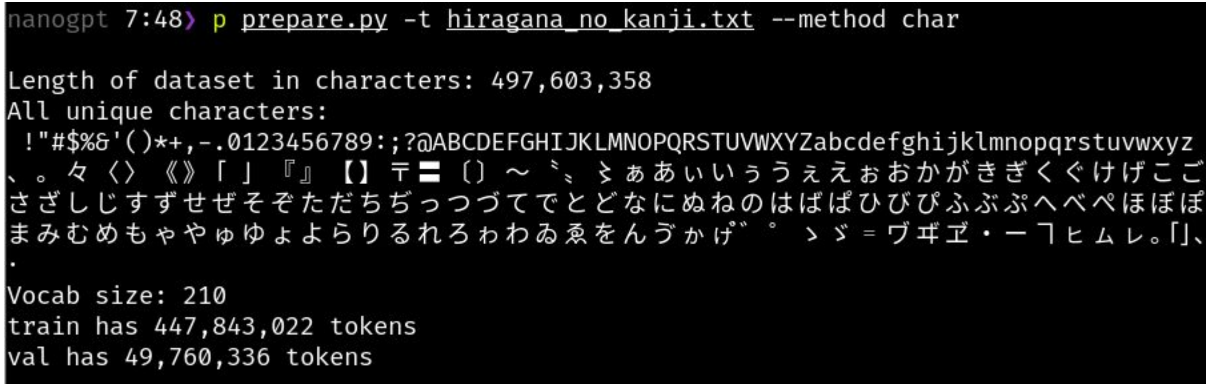
```
次の文を英語に翻訳してください。
最初の月面着陸は1969年7月20日のアポロ計画によって達成された
the first lunar landing was achieved by the Apollo program on July 20, 1969
```

- Preprocess Kanji -> Hiragana



```
nanogpt 23:20❯ p hiragana_converter.py japanese_eigo_kanji_hiragana_katakana.txt hiragana.txt
Converting:  15%|███                                      | 4151774/27894727 [01:58<11:21, 34817.19lines/s]
```

- Obtain ~210 Vocab Size Set



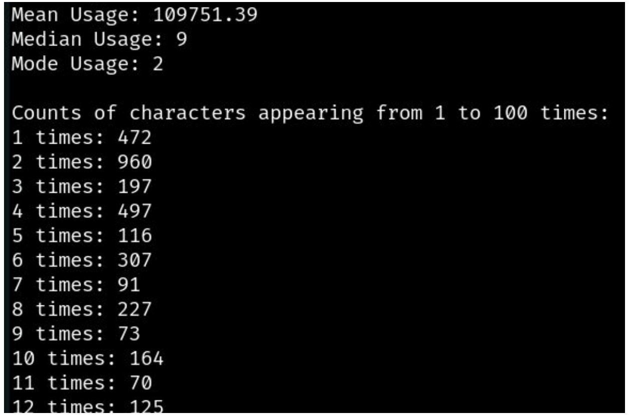
```
nanogpt 7:48❯ p prepare.py -t hiragana_no_kanji.txt --method char

Length of dataset in characters: 497,603,358
All unique characters:
 !"#$%&'()*+,-.0123456789:;?@ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
、。々〈〉《》「」『』【】〒〓〔〕～〝〟〵ぁあぃいぅうぇえぉおかがきぎくぐけげこご
さざしじすずせぜそぞただちぢっつづてでとどなにぬねのはばぱひびぴふぶぷへべぺほぼぽ
まみむめもゃやゅゆょよらりるれろゎわゐゑをんゔゕゖ゛゜ゝゞ゠ヷヸヹ・ーㇷㇶㇺㇾ｡｢｣､
･
Vocab size: 210
train has 447,843,022 tokens
val has 49,760,336 tokens
```

Long Tail Stats Before and After Kanji -> Hiragana

- Initially many characters appeared a small number of times

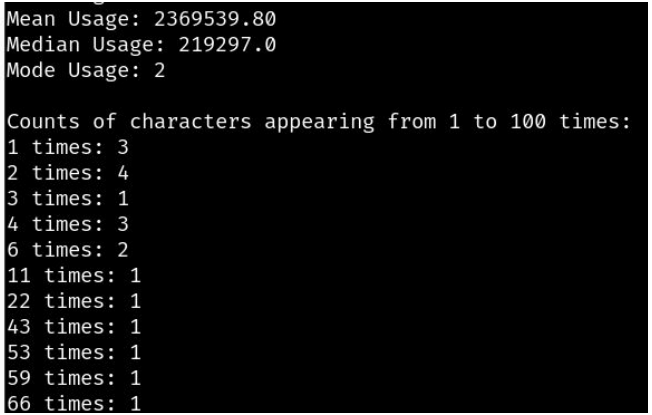
- Key Stats:
  - Before:
    - 7700 vocab
    - **Median = 9**
  - After Sub-Char:
    - 210 vocab
    - **Median = 219,297**

Before



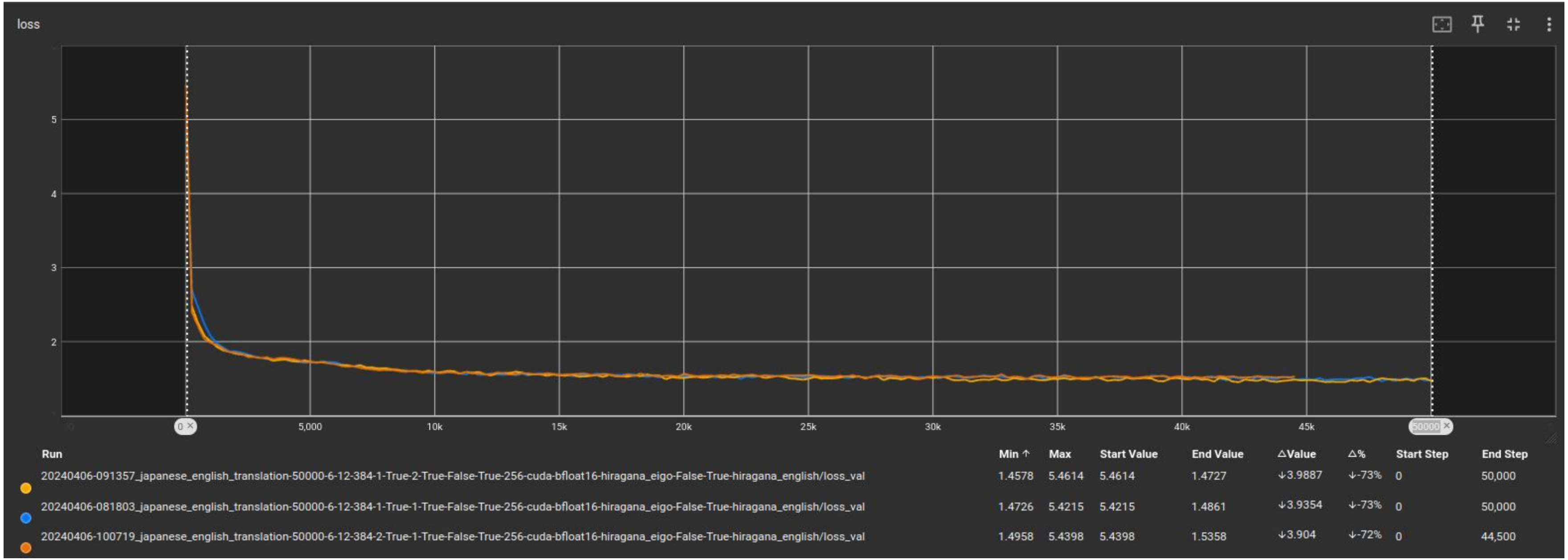
```
Mean Usage: 109751.39
Median Usage: 9
Mode Usage: 2

Counts of characters appearing from 1 to 100 times:
1 times: 472
2 times: 960
3 times: 197
4 times: 497
5 times: 116
6 times: 307
7 times: 91
8 times: 227
9 times: 73
10 times: 164
11 times: 70
12 times: 125
```

After



```
Mean Usage: 2369539.80
Median Usage: 219297.0
Mode Usage: 2

Counts of characters appearing from 1 to 100 times:
1 times: 3
2 times: 4
3 times: 1
4 times: 3
6 times: 2
11 times: 1
22 times: 1
43 times: 1
53 times: 1
59 times: 1
66 times: 1
```

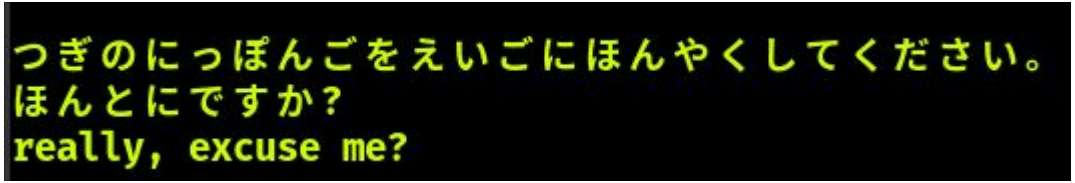
~1 Hour of Training Later… Will it Translate?



	Run	Min ↑	Max	Start Value	End Value	△Value	△%	Start Step	End Step
●	20240406-091357_japanese_english_translation-50000-6-12-384-1-True-2-True-False-True-256-cuda-bfloat16-hiragana_eigo-False-True-hiragana_english/loss_val	1.4578	5.4614	5.4614	1.4727	↓3.9887	↓-73%	0	50,000
●	20240406-081803_japanese_english_translation-50000-6-12-384-1-True-1-True-False-True-256-cuda-bfloat16-hiragana_eigo-False-True-hiragana_english/loss_val	1.4726	5.4215	5.4215	1.4861	↓3.9354	↓-73%	0	50,000
●	20240406-100719_japanese_english_translation-50000-6-12-384-2-True-1-True-False-True-256-cuda-bfloat16-hiragana_eigo-False-True-hiragana_english/loss_val	1.4958	5.4398	5.4398	1.5358	↓3.904	↓-72%	0	44,500

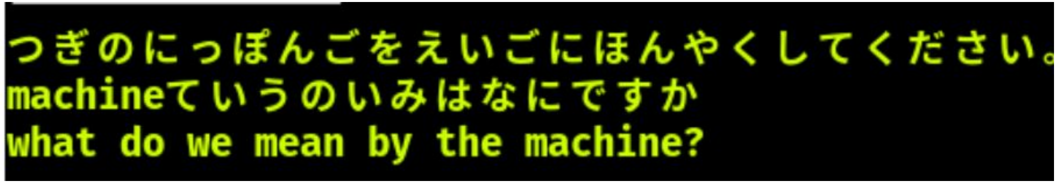
Spelling, Punctuation and Word Re-Ordering Demonstrated

- Learned many phrase translations



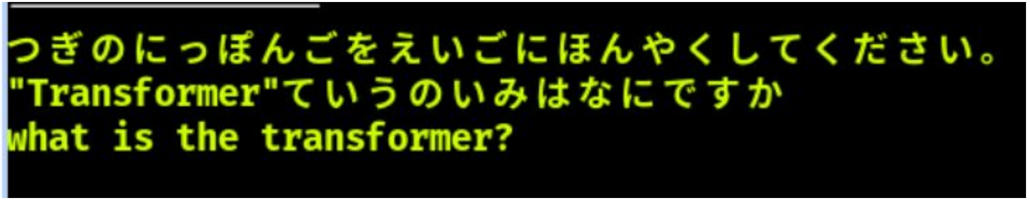
```
つぎのにっぽんごをえいごにほんやくしてください。
ほんとにですか?
really, excuse me?
```

- Observed to do re-ordering of out of distribution entities.



```
つぎのにっぽんごをえいごにほんやくしてください。
machineていうのいみはなにですか
what do we mean by the machine?
```

- *Above expectations for 1 hour training on 11M params*



```
つぎのにっぽんごをえいごにほんやくしてください。
"Transformer"ていうのいみはなにですか
what is the transformer?
```

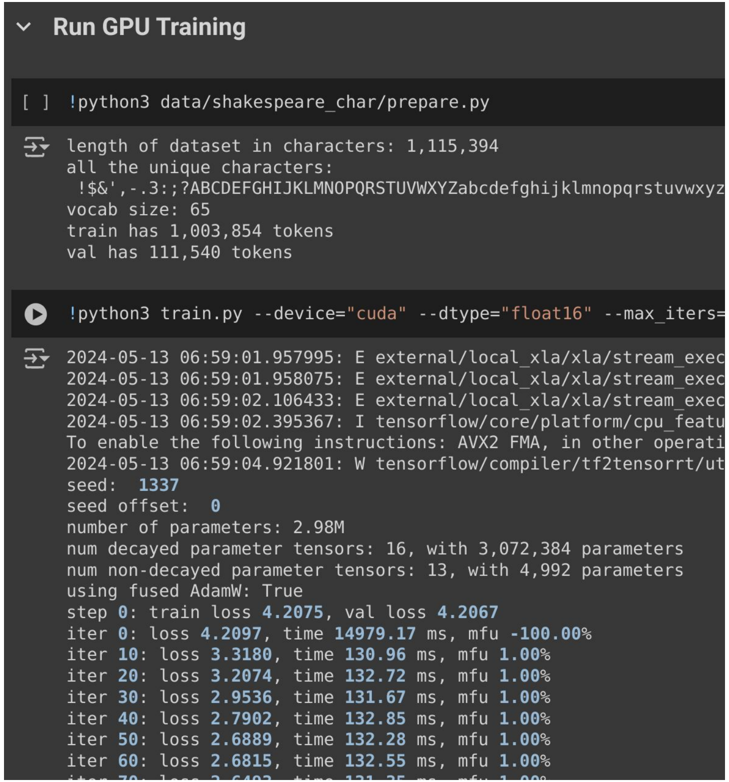
Colab

Workshop Colab

- Workshop colab
    - based on the popular NanoGPT framework

  https://colab.research.google.com/drive/1wfJB4k5KPUgsb0SIbyKtbjlXzBs4sMyI?usp=sharing#scrollTo=tUDCG38Ylpor

- We'll migrate to the Colab for the remainder of the workshop.



Run GPU Training

```
[ ] !python3 data/shakespeare_char/prepare.py
```

```
length of dataset in characters: 1,115,394
all the unique characters:
 !$&',-.3:;?ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
vocab size: 65
train has 1,003,854 tokens
val has 111,540 tokens
```

```
!python3 train.py --device="cuda" --dtype="float16" --max_iters=
```

```
2024-05-13 06:59:01.957995: E external/local_xla/xla/stream_exec
2024-05-13 06:59:01.958075: E external/local_xla/xla/stream_exec
2024-05-13 06:59:02.106433: E external/local_xla/xla/stream_exec
2024-05-13 06:59:02.395367: I tensorflow/core/platform/cpu_featu
To enable the following instructions: AVX2 FMA, in other operati
2024-05-13 06:59:04.921801: W tensorflow/compiler/tf2tensorrt/ut
seed:  1337
seed offset:  0
number of parameters: 2.98M
num decayed parameter tensors: 16, with 3,072,384 parameters
num non-decayed parameter tensors: 13, with 4,992 parameters
using fused AdamW: True
step 0: train loss 4.2075, val loss 4.2067
iter 0: loss 4.2097, time 14979.17 ms, mfu -100.00%
iter 10: loss 3.3180, time 130.96 ms, mfu 1.00%
iter 20: loss 3.2074, time 132.72 ms, mfu 1.00%
iter 30: loss 2.9536, time 131.67 ms, mfu 1.00%
iter 40: loss 2.7902, time 132.85 ms, mfu 1.00%
iter 50: loss 2.6889, time 132.28 ms, mfu 1.00%
iter 60: loss 2.6815, time 132.55 ms, mfu 1.00%
iter 70: loss 2.6403, time 131.35 ms, mfu 1.00%
```