COMP4901B: Large Language Models

Assignment 1 Report

HE, Wenqian
Student ID: 20860896

October 7, 2025

1 Part 1: Data Preprocessing

1.1 Cleaning Logic

For each paragraph in the text, I do the following cleaning steps:

- if there are at least 101 consecutive alphanumeric characters, I remove the paragraph
- if the paragraph has no punctuation, I remove the paragraph

1.2 Heuristic Quality Filter Logic

I reject the text if:

- there are any bad words in the text
- the text has no punctuation
- the text has no non-whitespace or punctuation characters
- the ratio of alphanumeric, punctuation, or whitespace characters to the total number of characters is less than 0.8

1.3 English Text Detection Logic

First, I filter the alphabetic characters, such as "a", "B", "我", etc., excluding the symbols like "?", "/", "-", etc.
Then, from the remaining alphabetic characters, I count the number of English alphabetic characters (i.e. "a-zA-Z").
If the ratio of English alphabetic characters to the total number of alphabetic characters is greater than the English character ratio threshold (I use 0.99 as the threshold), I accept the text.

1.4 Results

Here is the record numbers of WARC records processed and the number that pass all filters:

```
python homework.py --fname data.warc --output cleaned_test.txt --dfname
    topic_dataset.json --num_records 2000

704 passed out of 2000 records processed.
Cleaned documents saved to: cleaned_test.txt
121 deduplicated out of 219 records processed.
```

2 Part 2: Pretraining

The model is trained on Mac book with mps device enabled, which is achieved by modifying the `run_llama.py` file. This training takes about 9 hours.
Here is the training command:

```
python run_llama.py \
  --run_name run7 \
  --option pretrain \
  --data_path train_100M \
  --block_size  256 \
  --batch_size 512 \
  --micro_batch_size 32 \
  --epochs 1 \
  --tokenized_dir train_100M/tokenized \
  --use_gpu  \
  --val_path dev \
  --val_tokenized_dir dev/tokenized \
  --val_per_steps 200 \
  --test_path  test \
  --test_tokenized_dir test/tokenized \
  --auto_resume \
  --warmup_ratio 0.1 \
  --lr 1e-3
```

The training loss keep declining stably.
Here are some important curves:

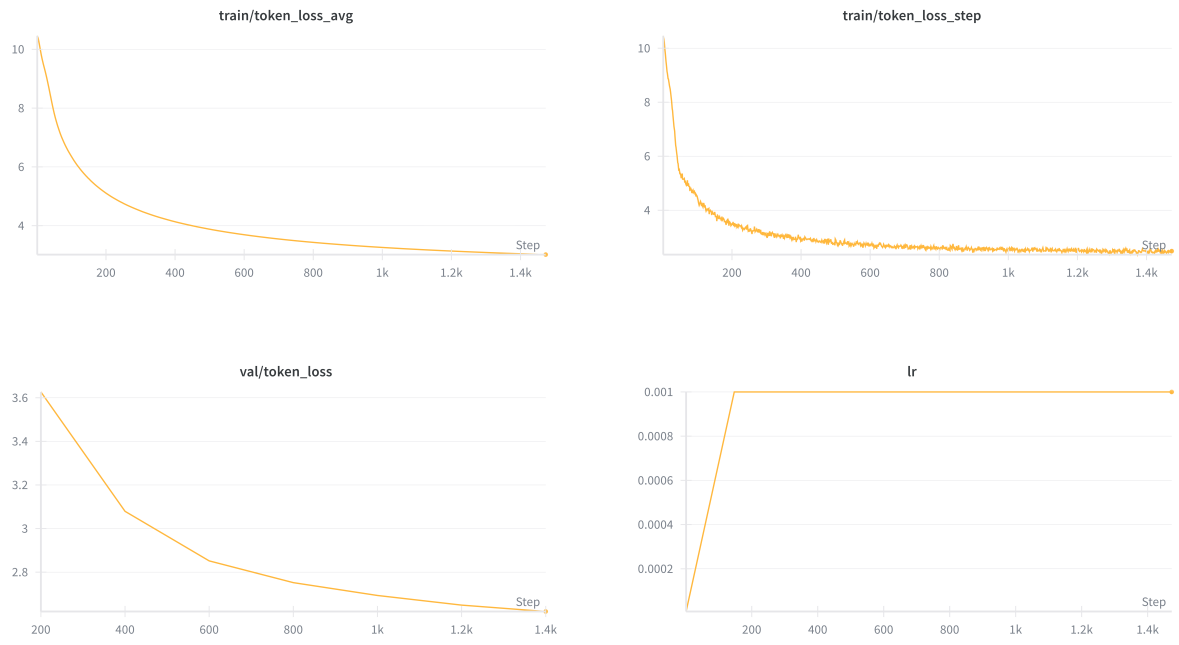

Figure 1: Training metrics visualization

3 Part 3: Generation

Provided model:

```
python run_llama.py --pretrained-model-path llama2-42M-babylm.pt --
    option generate

// temperature = 0.0
White Bird is a 2023 American war drama movie starring Diana Hunt,
    Diana Hunt, Diana Hunt, Diana Hunt, Diana Hunt, Diana Hunt, Diana
    Hunt, Diana Hunt, Diana Hunt, Diana Hunt, Diana Hunt, Diana Hunt,
    Diana Hunt, Diana Hunt, Diana Hunt,

// temperature = 0.5
White Bird is a 2023 American war drama movie starring James Dee, Eddie
     Quinn, Larry Dee, James McDonald, George J. L. Smith, David Dee,
    John Duck, James McDonald, John H. W. Bush. It was distributed by 20
    th Century Fox. = = = PG4021 = = = A LITTLE
```

My pretrained model:

```
python run_llama.py --pretrained-model-path run7-pretrain-1-0.001.pt --
    option generate

// temperature = 0.0
White Bird is a 2023 American war drama movie starring John H. W. Bush,
     John H. W. Bush, John H. W. Bush, John H. W. Bush, John H. W. Bush,
     John H. W. Bush, John H. W. Bush, John H. W. Bush, John H. W. Bush,
     John H. W. Bush, John H. W.

// temperature = 0.5
White Bird is a 2023 American war drama movie starring Shon Alley and
    Kyle, Marvin, John H. Sharine, Danny Reed, David H. W. Bush, J. W.
    Bush, David H. W. Bush, David H. Bush, Daniel H. Bush, David H. W.
    Bush, Michael Jackson, David K. Bush, David H. Bush, James K
```

From the generation results, although using temperature = 0.0 is more coherent, such as the generated sentence from my pretrained model, which echos "John H. W. Bush" again and again, I think the temperature = 0.5 is better. Using greedy method will lead to very repetitive and boring generation, as shown in the generated sentence from the both models. Using temperature sampling will lead to more diverse and interesting generation.