Homework 5

Anshul Sawant

February 16, 2025

Contents

1 Problem 1.1: Two Datasets

1.1 Dolma

1.1.1 A: Knowledge cutoff

Depending on source varies from Apr 2019 to Oct 2023. Overall cutoff is Oct 2023.

1.1.2 B: Data source

Mix of web content, academic publications, code, books and encyclopedic materials.

1.1.3 C: Model trained on the dataset

OPT

1.1.4 D: Dataset license

ODC-By. Different parts of dataset has different copyright protections. E.g., Arxiv (academic publications, most published under Creative Commons) vs Common crawl (web crawl, variable and/or unclear/unspecified copyrighting)

1.1.5 E: Task where trained model will do poorly

1. Tasks requiring up-to-date information.

2. Reasoning tasks.

1.2 Project Gutenberg

1.2.1 A: Knowledge cutoff

Unclear, whatever was the publication date of the latest book. Most of the content would be decades old.

1.2.2 B: Data source

Uncopyrighted (in the US) books and books explicitly allowed by authors for use by the project.

1.2.3 C: Model trained on the dataset

It is part of Dolma. Hence any models trained on Dolma. It is probably a part of many other datasets as well.

1.2.4 D: Dataset license

Uncopyrighted work can be distributed freely by anyone. For works where authors grant permission to the project, redistribution is restricted. Does LLM's output count as a redistribution? Debatable.

1.2.5 E: Task where trained model will do poorly

1. Generating a blog post. The tone of blogs is very different from that of most books.

2 Problem 1.2: Three Models

2.1 A: Three models chosen

GPT-2, Llama-2, OPT

2.2 B: Data training and processing

2.2.1 GPT2

1. **Dataset** They scraped all the web pages from outbound links on Reddit which received at least 3 karma. Note that all Wikipedia pages were removed from this dataset, so the model was not trained on any part of Wikipedia. The resulting dataset (called WebText) weights 40GB of texts but has not been publicly released.

2. **Processing** BPE with 50257 vocab size. Batch size of 1024 tokens.

2.2.2 LLaMA 2

1. **Dataset** Llama 2 was pretrained on 2 trillion tokens of data from publicly available sources. The fine-tuning data includes publicly available instruction datasets, as well as over one million new human-annotated examples. Neither the pretraining nor the fine-tuning datasets include Meta user data.

2. **Processing** Could find no information on this.

2.2.3 OPT

1. **Dataset** Consists of BookCorpus, CC-Stories, The Pile, Pushshift.io, CCNewsV2

2. **Processing**

   > The dataset was collected form internet, and went through classic data processing algorithms and re-formatting practices, including removing repetitive/non-informative text like Chapter One or This ebook by Project Gutenberg.

   GPT-2 BPE, batch size of 2048 tokens.

2.3 C: A use case for LLM trained on publicly accessible data

1. Publicly available datasets are often free or low-cost.

2. Research uses, such as ablation studies (for outcomes such as bias etc.)

3 Problem 2.1

3.1 A: Number of HTML pages

6368

3.2 B: Code and inline HTML

It parses it as is, retaining the white space formatting.

3.3 B: How does it handle HTML tags

Pretty much removes all formatting tags (headings, tables, paragraphs etc). Ignores images.

3.4 C: WET vs Cleaned HTML

The most significant differences I see is that

1. My `html_to_text` filters out non-roman alphabet languages.
2. `html_to_text` has very permissive puntuation set. Therefore a lot of inline characters like ##, * etc make it into text. WET version is probably better because of more restrictive punctuation set. It may be better for multi-lingual training as well. But that depends on the use case.

4 Problem 2.2

4.1 A: Documents Deleted

2572, 40% considered low quality

4.2 B: Low quality docs that passed the filter

1. `http://18ha.e11.tw/tag/869` It is mostly just URLs. Maybe lines that are just URLs must be filtered out.
2. `http://101lab.net/blog/2004/06/post-276.html` This is mostly chinese characters intersperced with dates and []. Maybe have a more restrictive punctuation set, fiter out links and have some assertion on distribution of alphabets and numbers in a paragraph.

4.3 C: Non-english languages

My filter tries to exclude all texts not in roman script.

4.4 D: Domain specific filtering

Coding domain will have different cleaning and fitering requirements. Depending on use case, we may want to remove comments. Filtering will certainly involve considering files names with certain extensions only.

4.5 E: Additional Data Filtering Stages

1. Language filtering (natural or coding)
2. Classifier based quality filtering
3. Deduplication (approximate or exact matching)
4. Domain specific cleaning, possibly based on different word distributions.

5 Problem 2.3

5.1 A: How long in seconds does it take to load the dataset?

Around 270 seconds. Processing around 25 documents per second. For 3 billion documents this should take around 10^8 seconds or around 3 years.

5.2 B: How to make it faster.

2.1 Parallelize in one machine using threads 2.2 Parallelize across machiens using Flume like frameworks 2.3 Minimize processing done in Python. Instead use libraries where the core functionality is implemented in C and that can process entire document as a unit.

5.3 C: Advantage of using packing over padding

One advantage is to minimize waste of training iterations on padded tokens. This is especially relevant for large models.