

- “The Pile: An 800GB Dataset of Diverse Text for Language Modeling” (2020) by Gao et al., <https://arxiv.org/abs/2101.00027>
- “The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only,” (2023) by Penedo et al., <https://arxiv.org/abs/2306.01116>
- “RedPajama,” by Together AI, <https://mng.bz/d6nw>
- The FineWeb Dataset, which includes more than 15 trillion tokens of cleaned and deduplicated English web data sourced from CommonCrawl, <https://mng.bz/rVzy>

The paper that originally introduced top-k sampling is

- “Hierarchical Neural Story Generation” (2018) by Fan et al., <https://arxiv.org/abs/1805.04833>

An alternative to top-k sampling is top-p sampling (not covered in chapter 5), which selects from the smallest set of top tokens whose cumulative probability exceeds a threshold p , while top-k sampling picks from the top k tokens by probability:

- Top-p sampling, https://en.wikipedia.org/wiki/Top-p_sampling

Beam search (not covered in chapter 5) is an alternative decoding algorithm that generates output sequences by keeping only the top-scoring partial sequences at each step to balance efficiency and quality:

- “Diverse Beam Search: Decoding Diverse Solutions from Neural Sequence Models” (2016) by Vijayakumar et al., <https://arxiv.org/abs/1610.02424>

Chapter 6

Additional resources that discuss the different types of fine-tuning are

- “Using and Finetuning Pretrained Transformers,” <https://mng.bz/VxJG>
- “Finetuning Large Language Models,” <https://mng.bz/x28X>

Additional experiments, including a comparison of fine-tuning the first output token versus the last output token, can be found in the supplementary code material on GitHub:

- Additional spam classification experiments, <https://mng.bz/AdJx>

For a binary classification task, such as spam classification, it is technically possible to use only a single output node instead of two output nodes, as I discuss in the following article:

- “Losses Learned—Optimizing Negative Log-Likelihood and Cross-Entropy in PyTorch,” <https://mng.bz/ZEJA>

You can find additional experiments on fine-tuning different layers of an LLM in the following article, which shows that fine-tuning the last transformer block, in addition to the output layer, improves the predictive performance substantially:

- “Finetuning Large Language Models,” <https://mng.bz/RZJv>

Readers can find additional resources and information for dealing with imbalanced classification datasets in the imbalanced-learn documentation:

- “Imbalanced-Learn User Guide,” <https://mng.bz/2KNa>

For readers interested in classifying spam emails rather than spam text messages, the following resource provides a large email spam classification dataset in a convenient CSV format similar to the dataset format used in chapter 6:

- Email Spam Classification Dataset, <https://mng.bz/1GEq>

GPT-2 is a model based on the decoder module of the transformer architecture, and its primary purpose is to generate new text. As an alternative, encoder-based models such as BERT and RoBERTa can be effective for classification tasks:

- “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding” (2018) by Devlin et al., <https://arxiv.org/abs/1810.04805>
- “RoBERTa: A Robustly Optimized BERT Pretraining Approach” (2019) by Liu et al., <https://arxiv.org/abs/1907.11692>
- “Additional Experiments Classifying the Sentiment of 50k IMDB Movie Reviews,” <https://mng.bz/PZJR>

Recent papers are showing that the classification performance can be further improved by removing the causal mask during classification fine-tuning alongside other modifications:

- “Label Supervised LLaMA Finetuning” (2023) by Li et al., <https://arxiv.org/abs/2310.01208>
- “LLM2Vec: Large Language Models Are Secretly Powerful Text Encoders” (2024) by BehnamGhader et al., <https://arxiv.org/abs/2404.05961>

Chapter 7

The Alpaca dataset for instruction fine-tuning contains 52,000 instruction–response pairs and is one of the first and most popular publicly available datasets for instruction fine-tuning:

- “Stanford Alpaca: An Instruction-Following Llama Model,” https://github.com/tatsu-lab/stanford_alpaca

Additional publicly accessible datasets suitable for instruction fine-tuning include

- LIMA, <https://huggingface.co/datasets/GAIR/lima>
 - For more information, see “LIMA: Less Is More for Alignment,” Zhou et al., <https://arxiv.org/abs/2305.11206>

- UltraChat, <https://huggingface.co/datasets/openchat/ultrachat-sharegpt>
 - A large-scale dataset consisting of 805,000 instruction-response pairs; for more information, see “Enhancing Chat Language Models by Scaling High-quality Instructional Conversations,” by Ding et al., <https://arxiv.org/abs/2305.14233>
- Alpaca GPT4, <https://mng.bz/Aa0p>
 - An Alpaca-like dataset with 52,000 instruction-response pairs generated with GPT-4 instead of GPT-3.5

Phi-3 is a 3.8-billion-parameter model with an instruction-fine-tuned variant that is reported to be comparable to much larger proprietary models, such as GPT-3.5:

- “Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone” (2024) by Abdin et al., <https://arxiv.org/abs/2404.14219>

Researchers propose a synthetic instruction data generation method that generates 300,000 high-quality instruction-response pairs from an instruction fine-tuned Llama-3 model. A pretrained Llama 3 base model fine-tuned on these instruction examples performs comparably to the original instruction fine-tuned Llama-3 model:

- “Magpie: Alignment Data Synthesis from Scratch by Prompting Aligned LLMs with Nothing” (2024) by Xu et al., <https://arxiv.org/abs/2406.08464>

Research has shown that not masking the instructions and inputs in instruction fine-tuning effectively improves performance on various NLP tasks and open-ended generation benchmarks, particularly when trained on datasets with lengthy instructions and brief outputs or when using a small number of training examples:

- “Instruction Tuning with Loss Over Instructions” (2024) by Shi, <https://arxiv.org/abs/2405.14394>

Prometheus and PHUDGE are openly available LLMs that match GPT-4 in evaluating long-form responses with customizable criteria. We don’t use these because at the time of this writing, they are not supported by Ollama and thus cannot be executed efficiently on a laptop:

- “Prometheus: Inducing Finegrained Evaluation Capability in Language Models” (2023) by Kim et al., <https://arxiv.org/abs/2310.08491>
- “PHUDGE: Phi-3 as Scalable Judge” (2024) by Deshwal and Chawla, “<https://arxiv.org/abs/2405.08029>
- “Prometheus 2: An Open Source Language Model Specialized in Evaluating Other Language Models” (2024), by Kim et al., <https://arxiv.org/abs/2405.01535>

The results in the following report support the view that large language models primarily acquire factual knowledge during pretraining and that fine-tuning mainly enhances their efficiency in using this knowledge. Furthermore, this study explores