GPT-2 Metadata Pretraining Towards Instruction Finetuning for Ukrainian

Volodymyr Kyrylov
Università della Svizzera italiana
vol@wilab.org.ua

Dmytro Chaplynskyi
lang-uk
chaplinsky.dmitry@gmail.com

Abstract

We explore pretraining unidirectional language models on 4B tokens from the largest curated corpus of Ukrainian, UberText 2.0. We enrich document text by surrounding it with weakly structured metadata, such as title, tags, and publication year, enabling metadata-conditioned text generation and text-conditioned metadata prediction at the same time. We pretrain GPT-2 Small, Medium, and Large models on a single GPU, reporting training times, BPC on BrUK, BERTScore, and BLEURT on titles for 1000 News from the Future. Next, we venture to formatting POS and NER datasets as instructions, and train low-rank attention adapters, performing these tasks as constrained text generation. We release our models for the community at https://github.com/proger/uk4b.

1 Introduction

Large language models provide a text-based user interface to perform multiple language processing tasks. The release of UberText 2.0 (Chaplynskyi, 2023) is a milestone that unlocks pretraining experiments of language models on curated Ukrainian texts. Coupled with recent improvements to hardware and software, we can train larger models on a single consumer GPU from scratch.

Our contributions are:

- techniques to train language models on UberText 2.0 under 1B parameters on consumer hardware setting a baseline of 1.43 BPC on a subset of BrUK;
- a method to add new tasks from document metadata in pretraining compared to finetuning larger models for sequence generation explicitly;
- exploration of tagging tasks formatted as instructions using low-rank adapters compared to traditional sequence tagging methods.

2 Related Work

Radford and Narasimhan (2018) show that a single pretrained causal Transformer (Vaswani et al., 2017) decoder-only model on as much as 5 GB of books with 124M parameters can be finetuned for many downstream tasks. Devlin et al. (2019) show that using an bidirectional encoder-only model improves performance for tasks where bidirectional context is important, like question answering. Radford et al. (2019) discover that models pretrained on 40 GB of curated internet text and scaled up to over 1B parameters are able to perform multiple tasks in zero shot scenario. 100x larger models trained on larger dataset exhibit few shot learning abilities of new tasks at the cost of impressive engineering efforts (Brown et al., 2020; Chowdhery et al., 2022). These ideas guide us towards seeking large text corpora and training Transformers on them.

Kaplan et al. (2020) and Hoffmann et al. (2022) observe that bigger models converge to the same validation loss much faster in the same wall clock time. They fit a power law curve between a power of the model size, dataset size, or compute time and performance ($l = ax^{b<1} + c$) into runtime metrics collected from running a large number of experiments. The power laws suggest that the returns from increasing model, data, or compute diminish after a certain point. Caballero et al. (2022) present a smoothly broken neural scaling law equation, suggesting a scaling speedup laying further ahead past the currently accepted inflection region. Sorscher et al. (2022) suggest a way to beat scaling laws by using careful data selection methods on vision tasks. These ideas give us the insight that we should use the biggest models possible for our compute budget.

It's not only compute that's important. While the work of Radford et al. (2019) discovered prompts that drove the model to perform tasks like summarization, Schick and Schütze (2021) introduce pattern-exploiting training that reformulates sentences into cloze tasks on purpose. It is beneficial to curate examples of natural language instructions to save compute.

Instruction finetuning datasets, such as The Flan Collection, released by Longpre et al. (2023), curate massive amounts of task-specific datasets and provide a pipeline to reformulate tasks into natural language using seqio introduced in Roberts et al. (2022). Flan T5 demonstrates that you can achieve higher performance on multiple NLP tasks at once with smaller models in 1.5B–11B range using such data curation methods. These ideas inspire us to leverage metadata and attempt to formulate NLP tasks using natural language.

Techniques like sequence length warmup (Li et al., 2022), gradient clipping (Graves, 2013) enable training stability. Dettmers et al. (2022) enable memory savings by quantizing gradient statistics. Katharopoulos et al. (2020) explore a recurrent formulation of attention with lower computational complexity, and Schlag et al. (2021) view it as fast weight programmers improving capacity of attention in the recurrent setting. Tillet et al. (2019) provide a programming language to implement high performing kernels quickly. Dao et al. (2022) demonstrate how to significantly speed up computation of self-attention and allow much larger context sizes than 1024 or 2048 tokens. Finally, Geiping and Goldstein (2022) demonstrate achieving competitive pretraining speed and performance on a single GPU in 24 hours with a BERT-like model. Notably, these two advancements, the release of PyTorch 2.0 and Andrej Karpathy's nanoGPT tweets, encouraged us to try pretraining from scratch.

Low-rank adaptation methods presented in Hu et al. (2022) and extended in Valipour et al. (2022) enable finetuning of large pretrained models on consumer hardware by updating only a small fraction of extra parameters, suggesting we can efficiently maintain adapters for many tasks in memory at once and achieve better finetuning performance.

Shen et al. (2022) observe that smaller models optimize faster in the beginning of training and propose grafting parameters of a smaller network onto a larger one to continue training after some time. We keep this idea in mind for the future.

3 Pretraining

3.1 Dataset Preparation

We produce a tokenizer from the Wikipedia subset of the corpus using SentencePiece (Kudo and Richardson, 2018) on the document level, including whitespace symbols like newlines and byte-level fallback, totaling 50257 tokens[1]. We include additional special tokens, like `<|transcribe|>`, reserved for future use. Every document is Unicode-normalized using ftfy[2]. We tokenize the News, Fiction and Wikipedia subsets of UberText 2.0 in parallel using Datasets (Lhoest et al., 2021).

When tokenizing each document we prepend `title`, year part of `date_of_publish` and `tags` document metadata fields prefixed by тема: ("topic: "), рік: ("year: ") and мітки: ("tags: ") strings in randomized order, separated by newlines from each other, and by double newlines from the body. The metadata is repeated at the end of the document as well after a double newline. After the metadata suffix we append one `<|endoftext|>` token. Following Geiping and Goldstein (2022) we remove all documents that have a ratio of characters to tokens higher than 0.4.

The resulting dataset has 4,299,910,622 training tokens. 4,194,956 tokens are set aside for validation. All document tokens are concatenated together into a single binary file with 2 bytes per token. We name this dataset `uk4b` in our experiments.

3.2 Model

We choose a Transformer decoder based on GPT-2 (Radford et al., 2019). The decoder contains two embedding tables, one for each of 50257 tokens and one for each of 1024 possible token positions. At input, every token in a sequence is represented using a sum of the token embedding and its corresponding position embedding. Input goes through N blocks, consisting of a residually connected multi head self-attention layer, followed by layer normalization and a residually connected linear layer, followed by another layer normalization. Latent representation is projected back to token ids using a linear layer with weights tied to the token embedding table.

Attention heads are constrained to use only tokens earlier in a sequence. This enables us to use an

[1] Original GPT-2 uses 50000 BPE tokens + 256 for each byte + 1 for `<|endoftext|>`

[2] `https://ftfy.readthedocs.io`

Model	Size	**BrUK$_{29k}$**	**uk4b validation**	**uk4b training**	**ETA**
		bpc↓	loss↓	tokens (compute optimal)	3090-hours
LSTM	5.7M	1.71	-	-	-
GPT-2 Small$_+$	123M	1.50	2.38	6.87B (2.29B)	35
GPT-2 Medium$_+$	355M	1.46	2.10	6.29B (6.85B)	89
GPT-2 Large$_l$	774M	**1.43**	1.82	21.4B (15.4B)	492$_\dagger$

Table 1: Intrinsic evaluation of trained models. $_+$ means the model uses an output projection layer with a dimension rounded up to the next multiple of 8 to enable tiling optimizations, and biases from all attention, linear and layer normalization layers have been removed. $\cdot_l$ means the model uses layer normalization of token and position embeddings. $\cdot_\dagger$ denotes that the time estimate for Large is computed for a 772M $_+$-type model with 2048 tokens per forward pass. LSTM is trained on a different train/validation split of UberText 2.0 than uk4b and is available at https://huggingface.co/lang-uk/flair-uk-forward.

autoregressive text completion objective computed in *parallel for all tokens* in a batch.

Our implementation is based on nanoGPT.[3] We rely on PyTorch (Paszke et al., 2019) 2.0 compiler and FlashAttention (Dao et al., 2022).

We pretrain three model variants: Small, Medium and Large.

Small has 12 layers, 12 attention heads and 768 embedding dimension totaling 124M parameters. We do not use the bias in attention, linear layer and layer normalization for speed. We use AdamW $\beta_1 = 0.9, \beta_2 = 0.95$, weight decay of 1e-2. Learning rate is linearly warmed up for 1000 steps from 6e-5 to 6e-4 and then back for 13000 more steps. We clip gradients at 2-norm of 1.

We use a batch size of 512 with sequence length 1024.

Medium has 24 layers, 16 attention heads and 1024 embedding dimension totaling 354M parameters, without bias. According to Chinchilla Approach 2 (Hoffmann et al., 2022)[4] compute optimal estimate we need to train on 6.85B tokens, requiring 13066 gradient updates. We round it up to 13100 updates. We train Medium and Small for the same amount of time to compare wall clock time on RTX 3090. Small and Medium vocabulary size is expanded to 50304 to enable tiling optimizations[5]

Large has 774M parameters: 36 layers, 20 attention heads and 1280 embedding dimension. We used bias in all layers in this model. We train Large for 10M forward passes on a single A100. Compute optimal estimate for Large is 15.4B tokens, requiring roughly 29.5K gradient updates. At 2048 tokens per iteration this requires 7.5M forward passes. The training was started with 8-bit AdamW (Dettmers et al., 2022) and continued with 32-bit AdamW following divergence. We used a maximum learning rate of 2.5e-4. As an artifact, this model additionally includes layer normalization in Embedding layers.

Large model uses standard PyTorch initialization for all layers, Small and Medium use GPT-2 initialization. We use bfloat16 adaptive mixed precision in all runs. Loss curves are available on Figure 1. Sequences of tokens are randomly sampled from the dataset during training.

One epoch of uk4b requires 8202 gradient updates. Compute optimal training tokens estimate assumes tokens are not repeated, which is not the case for our experiment.

3.3 Evaluation

To perform instrinsic evaluation, we use a subset of BrUK corpus of contemporary Ukrainian by Starko et al. (2016-2023). To avoid overlap with training data, we choose sentences split using a toolkit by Rysin (2022) that do not appear in UberText 2.0, ending up with 28643 test sentences. We call this dataset BrUK$_{29k}$. As a baseline, we include a character-level 1-layer LSTM (Hochreiter and Schmidhuber, 1997) with hidden size 1024 trained for 20 epochs (364B characters) on another variant of UberText 2.0 using an implementation provided by Akbik et al. (2018). We report bits per character and training statistics in Table 1 (Mielke, 2019).

3.4 Metadata Prediction

To evaluate metadata prediction, we sample 1000 News Articles from the Future using an in-domain news source.

We perform decoding of the Large model

[3] https://github.com/karpathy/nanoGPT

[4] Estimated using code from https://github.com/karpathy/nanoGPT/blob/master/scaling_laws.ipynb

[5] Once again thanks to @karpathy: https://twitter.com/karpathy/status/1621578354024677377

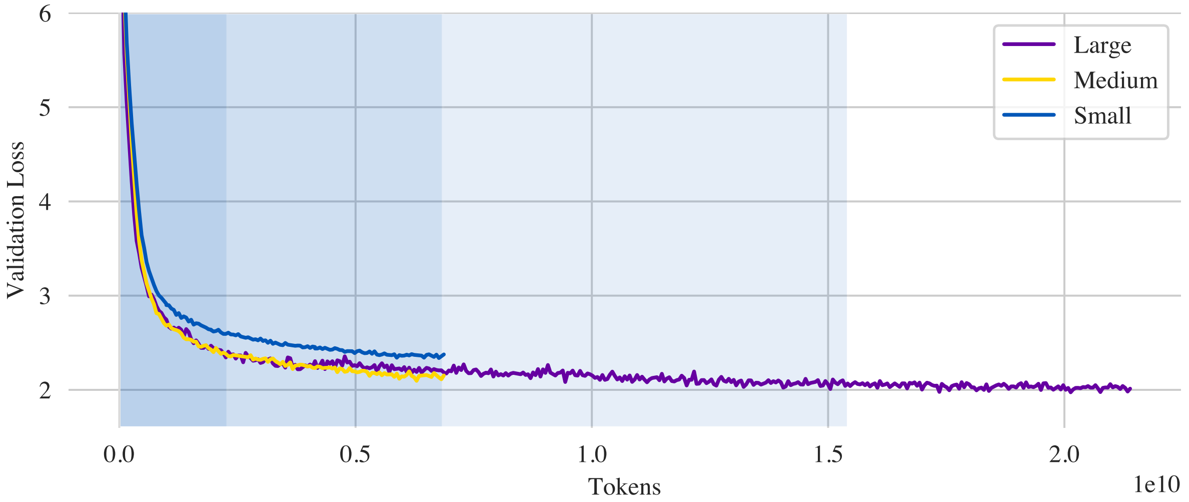

Figure 1: Validation loss curves against training tokens seen by models. Shaded regions denote compute optimal training times for Small, Medium and Large estimated using Chinchilla (Hoffmann et al., 2022).

prompted by article content followed by two newlines and prompt tokens тема: ("topic: ") or мітки: ("tags: ").

We report BERTScore using xlm-roberta-large (Zhang* et al., 2020) and BLEURT using BLEURT-20 model (Sellam et al., 2020; Pu et al., 2021) for title prediction task in Table 2. To compare, we take mBART-50 (Tang et al., 2021), which is an encoder-decoder model pretrained on multiple languages and finetune it on news articles from UberText 2.0. We remove all text from mBART output after the first sentence.

For tag prediction, we measure and report intersection over union and accuracy between sets of reference and hypothesis tags constructed by splitting the tag string by commas and downcasing.

Table 2: Metadata Prediction results on 1000 News Articles from the Future, Greedy Decoding. mBART is finetuned on 1000 news articles from UberText 2.0.

Titles	BERTScore F1	BLEURT mean
GPT-2 Small 123M	0.90	0.54
GPT-2 Medium 355M	0.91	0.57
GPT-2 Large 774M	0.91	0.59
mBART 610M	**0.94**	**0.74**
Tags	IOU	Accuracy
GPT-2 Small 123M	0.47	0.64
GPT-2 Medium 355M	0.54	0.71
GPT-2 Large 774M	0.56	0.71

4 Finetuning

4.1 Low-Rank Adaptation

When finetuning for a new task, we add low-rank decomposed clones of query W_q and key W_k input projection weights for each attention head, summing their activations with original queries and keys, as suggested by Hu et al. (2022) using their provided code. This method is based on an observation that overparametrized models reside in a low intrinsic dimension by Li et al. (2018). Practically, this allows us to finetune large models on consumer GPUs by updating only a small amount of parameters. The pretrained model remains frozen, allowing operation of multiple adaptation modules on a single GPU at once.

4.2 Instruction Datasets

Wei et al. (2021) has shown that finetuning large models on instruction datasets improves their zero-shot performance. In aspiration to this work, we prepare POS (Kotsyba et al., 2018) and lang-uk[6] NER datasets in instruction format to evaluate our model on these tasks in a finetuned setting.

For each example, we prefix the input sentence by a prompt token речення: ("sentence: "), provide the input sentence and put a task prompt проаналізуй: ("analyze: ") on a new line followed by a response. We format ground truth responses to contain observed words interspersed with hidden labels: part-of-speech tags in case of POS and named entity labels in case of NER. Word

[6] https://lang.org.ua

tokenization depends on the task, making the task harder than pointwise token projection as the model needs to learn arbitrary tokenization. We ensure hidden labels use exactly one token. We prompt the hidden label prompt by a / token. This encoding reminds us of a text representation of observed-hidden sequences in hidden Markov models.

We intercalate all examples with an <|endoftext|> training and continue training using the same objective using the same data loading process as during pretraining.

During our preliminary experiments, we observe that the model struggles to correctly reproduce the sentence after the prompt in about 1/3rd of the cases, making evaluation impossible without constrained decoding.

To complete POS measurements we provide the model with a oracle-tokenized observed response with hidden labels replaced by a token previously unseen during training[7]. We evaluate by forwarding this string through the model and replacing blanks with highest probability tokens. We effectively use an autoregressive model in a parallel fashion. We do not constrain the set of tokens to choose from after the forward pass. The results in the evaluation are available in Table 3.

Table 3: POS Performance

Model	Accuracy
Flair LSTM Forward/Backward	**0.979**
UDPipe	0.975
GPT-2 Medium Instr. Parallel **(ours)**	0.964
FastText CBOW (flair)	0.940
FastText CBOW (spacy)	0.825

To complete NER evaluations, we provide the model with oracle tokenization, performing constrained greedy decoding. Results of this evaluation are show in Table 4. ELECTRA models are provided by updated work of Schweter (2020).

5 Discussion

We are excited to release a new decoder-only monolingual model trained on curated Ukrainian data to the community.

It took us over a month to pretrain the first Large model successfully and in the process we became aware of possible improvements to the model, such as removing biases. These improvements resulted in a narrow visual gap between Medium and Large, as seen on Figure 1. We were able to report a much lower validation loss on Large due to a spike towards the end of training. Loss curves in Figure 1, bpc values in Table 1 and results on metadata prediction show in Table 2 suggest it might be beneficial to train Medium for longer. We see that using a task specific encoder-decoder model is performing better, possibly leveraging context in both directions when predicting metadata given the document.

Table 4: NER Performance

Model	F1	Prec	Recall
xlm-roberta-large	**0.92**	**0.92**	**0.91**
xlm-roberta-base	0.89	0.89	0.88
dbmdz/electra-base-ukrainian-cased-discriminator	0.89	0.89	0.89
lang-uk/electra-base-ukrainian-cased-discriminator	0.87	0.87	0.87
youscan/ukr-roberta-base	0.87	0.87	0.86
bert-base-multilingual-cased	0.87	0.88	0.87
Flair LSTM Forward and Backward	0.86	0.86	0.86
GPT-2 Large Instruction Data, Constrained Decoding **(ours)**	0.85	0.86	0.84
FastText CBOW	0.83	0.86	0.80
FastText skipgram	0.82	0.83	0.81

While aiming towards a general purpose language agent trained on a single GPU, we are lured by simplicity of formatting tasks as instructions. During our experiments, we observed the model drifting away from the NER task into text generation on long inputs, requiring us to use constrained decoding to "remind" it what the model is supposed to be doing. We achieve a competitive result this way, however would still choose a more traditional approach to solve NER, as confirmed by our measurements in Table 4.

It is suprising to find that POS could be solved by "filling in blanks" by picking maximum probability tokens in parallel. We used that result in Table 3.

There is room for more data to faithfully leverage the prediction of the number of tokens we need to train for to optimally utilize compute. There is more available data in Conneau et al. (2020), Wenzek et al. (2020) and Raffel et al. (2020). We leave filtering this data to future work.

Limitations

We choose to keep bias of UberText 2.0 in the models as is. We observe a gap in performance be-

[7] we choose _ at random

tween our models and task-specific large encoder or encoder-decoder models. While we evaluate document-conditional metadata generation, we do not evaluate metadata-conditioned document generation ability present in our model. Constrained decoding necessary for NER evaluation is a major limitation of our instruction finetuning attempts, suggesting we need to make further improvements to the design of our pretraining corpus for performing multiple tasks with one model. We do not test our model on traditional sequence tagging formulations of POS and NER. Causal language models are useful for tasks like speech recognition and we leave effectiveness of these models on such tasks to future work.

Ethics Statement

We seek to accelerate adoption of larger language models at scale enabling new capabilities for Ukrainian, improving lives of millions of language users. We recognize that our work can be misused to produce fake information and deceptive content and we do not condone such use of our models.

Acknowledgements

Volodymyr would like to thank Kazuki Irie for encouragement to train larger language models sooner and feedback on the early versions of the paper. Róbert Csordás and Aditya Ramesh for discussions about Transformers.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Ethan Caballero, Kshitij Gupta, Irina Rish, and David Krueger. 2022. Broken Neural Scaling Laws. In *NeurIPS ML Safety Workshop*.

Dmytro Chaplynskyi. 2023. Introducing UberText 2.0: a corpus of modern Ukrainian at scale. In *Proceedings of the Second Ukrainian Natural Language Processing Workshop*, pages 1–10, Dubrovnik, Croatia. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling language modeling with pathways.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Tri Dao, Daniel Y Fu, Stefano Ermon, Atri Rudra, and Christopher Re. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022. 8-bit optimizers via block-wise quantization. In *International Conference on Learning Representations*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jonas Geiping and Tom Goldstein. 2022. Cramming: Training a language model on a single gpu in one day. *ArXiv*, abs/2212.14034.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *ArXiv*, abs/1308.0850.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Tom Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karen Simonyan, Erich Elsen, Oriol Vinyals, Jack William Rae, and Laurent Sifre. 2022. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jared Kaplan, Sam McCandlish, T. J. Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeff Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *ArXiv*, abs/2001.08361.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are RNNs: Fast autoregressive transformers with linear attention. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5156–5165. PMLR.

Natalia Kotsyba, Bohdan Moskalevskyi, and Mykhailo Romanenko et al. 2018. Laboratorija ukrajins'koji.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A community library for natural language processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 175–184, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. 2018. Measuring the intrinsic dimension of objective landscapes. In *International Conference on Learning Representations*.

Conglong Li, Minjia Zhang, and Yuxiong He. 2022. The stability-efficiency dilemma: Investigating sequence length warmup for training GPT models. In *Advances in Neural Information Processing Systems*.

S. Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V. Le, Barret Zoph, Jason Wei, and Adam Roberts. 2023. The Flan Collection: Designing data and methods for effective instruction tuning. *ArXiv*, abs/2301.13688.

Sabrina J. Mielke. 2019. Can you compare perplexity across different segmentations?

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Amy Pu, Hyung Won Chung, Ankur Parikh, Sebastian Gehrmann, and Thibault Sellam. 2021. Learning compact metrics for MT. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 751–762, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Alec Radford and Karthik Narasimhan. 2018. Improving language understanding by generative pretraining.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research*, 21(140):1–67.

Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, Curtis Hawthorne, Aitor Lewkowycz, Alex Salcianu, Marc van Zee, Jacob Austin, Sebastian Goodman, Livio Baldini Soares, Haitang Hu, Sasha Tsvyashchenko, Aakanksha Chowdhery, Jasmijn Bastings, Jannis Bulian, Xavier Garcia, Jianmo Ni, Andrew Chen, Kathleen Kenealy, Jonathan H. Clark, Stephan Lee, Dan Garrette, James Lee-Thorp, Colin Raffel, Noam Shazeer, Marvin Ritter, Maarten Bosma, Alexandre Passos, Jeremy Maitin-Shepard, Noah Fiedel, Mark Omernick, Brennan Saeta, Ryan Sepassi, Alexander Spiridonov, Joshua Newlan, and Andrea Gesmundo. 2022. Scaling up models and data with t5x and seqio. *CoRR*, abs/2203.17189.

Andrij Rysin. 2022. nlp_uk: A collection of NLP tools and resources for Ukrainian language processing. `https://github.com/brown-uk/nlp_uk`. Accessed: February 18, 2023.

Timo Schick and Hinrich Schütze. 2021. It's not just size that matters: Small language models are also few-shot learners. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2339–2352, Online. Association for Computational Linguistics.

Imanol Schlag, Kazuki Irie, and Jürgen Schmidhuber. 2021. Linear transformers are secretly fast weight programmers. In *International Conference on Machine Learning*.

Stefan Schweter. 2020. Ukrainian ELECTRA model.

Thibault Sellam, Dipanjan Das, and Ankur Parikh. 2020. BLEURT: Learning robust metrics for text generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7881–7892, Online. Association for Computational Linguistics.

Sheng Shen, Pete Walsh, Kurt Keutzer, Jesse Dodge, Matthew Peters, and Iz Beltagy. 2022. Staged training for transformer language models. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 19893–19908. PMLR.

Ben Sorscher, Robert Geirhos, Shashank Shekhar, Surya Ganguli, and Ari S. Morcos. 2022. Beyond neural scaling laws: beating power law scaling via data pruning. In *Advances in Neural Information Processing Systems*.

Vasyl Starko, Andriy Rysin, Olha Havura, and Nataliia Cheilytko et al. 2016-2023. BRUK: Braunskyi korpus ukrainskoi movy.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2021. Multilingual translation from denoising pre-training. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3450–3466, Online. Association for Computational Linguistics.

Philippe Tillet, H. T. Kung, and David D. Cox. 2019. Triton: an intermediate language and compiler for tiled neural network computations. *Proceedings of the 3rd ACM SIGPLAN International Workshop on Machine Learning and Programming Languages*.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2022. DyLoRA: Parameter efficient tuning of pre-trained models using dynamic search-free low-rank adaptation. *ArXiv*, abs/2210.07558.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Thomas Wang, Adam Roberts, Daniel Hesslow, Teven Le Scao, Hyung Won Chung, Iz Beltagy, Julien Launay, and Colin Raffel. 2022. What language model architecture and pretraining objective work best for zero-shot generalization? In *International Conference on Machine Learning*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2021. Finetuned language models are zero-shot learners. *ArXiv*, abs/2109.01652.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Tianyi Zhang*, Varsha Kishore*, Felix Wu*, Kilian Q. Weinberger, and Yoav Artzi. 2020. BERTScore: Evaluating text generation with BERT. In *International Conference on Learning Representations*.

A Bidirectional Finetuning

Wang et al. (2022) presents experiments suggesting it's possible to perform MLM adaptation of causal models with only 1.3x of compute. We attempt to turn our Medium left-to-right model into a bidirectional one by relaxing the causal attention and continuing training the whole model using a masked language modeling objective for 4100 gradient updates. We observe a very sharp drop in the loss, converging to a degenerate solution: the model latches onto reproducing a single word instead of a blank token. We leave comprehensive evaluation of bidirectional adaptation of smaller models to future work.