It introduces additional self-supervised adversarial and controllable language modeling losses to the pre-training step, which enables ERNIE 3.0 Titan to beat other LLMs in their manually selected Factual QA task set evaluations.

GPT-NeoX-20B [118]: An auto-regressive model that largely follows GPT-3 with a few deviations in architecture design, trained on the Pile dataset without any data deduplication. GPT-NeoX has parallel attention and feed-forward layers in a transformer block, given in Eq. 4, that increases throughput by 15%. It uses rotary positional embedding [66], applying it to only 25% of embedding vector dimension as in [119]. This reduces the computation without performance degradation. As opposed to GPT-3, which uses dense and sparse layers, GPT-NeoX-20B uses only dense layers. The hyperparameter tuning at this scale is difficult; therefore, the model chooses hyperparameters from the method [6] and interpolates values between 13B and 175B models for the 20B model. The model training is distributed among GPUs using both tensor and pipeline parallelism.

$$x + Attn(LN_1(x)) + FF(LN_2(x)) \tag{4}$$

OPT [14]: It is a clone of GPT-3, developed to open-source a model that replicates GPT-3 performance. Training of OPT employs dynamic loss scaling [120] and restarts from an earlier checkpoint with a lower learning rate whenever loss divergence is observed. Overall, the performance of OPT-175B models is comparable to the GPT3-175B model.

BLOOM [13]: A causal decoder model trained on the ROOTS corpus to open-source an LLM. The architecture of BLOOM is shown in Figure 9, with differences like ALiBi positional embedding, an additional normalization layer after the embedding layer as suggested by the bitsandbytes[1] library. These changes stabilize training with improved downstream performance.

GLaM [91]: Generalist Language Model (GLaM) represents a family of language models using a sparsely activated decoder-only mixture-of-experts (MoE) structure [121, 90]. To gain more model capacity while reducing computation, the experts are sparsely activated where only the best two experts are used to process each input token. The largest GLaM model, GLaM (64B/64E), is about 7× larger than GPT-3 [6], while only part of the parameters are activated per input token. The largest GLaM (64B/64E) model achieves better overall results as compared to GPT-3 while consuming only one-third of GPT-3's training energy.

MT-NLG [117]: A 530B causal decoder based on the GPT-2 architecture that has roughly 3× GPT-3 model parameters. MT-NLG is trained on filtered high-quality data collected from various public datasets and blends various types of datasets in a single batch, which beats GPT-3 on several evaluations.

Chinchilla [96]: A causal decoder trained on the same dataset as the Gopher [116] but with a little different data sampling distribution (sampled from MassiveText). The model architecture is similar to the one used for Gopher, with the exception of AdamW optimizer instead of Adam. Chinchilla identifies the

[1] https://github.com/TimDettmers/bitsandbytes

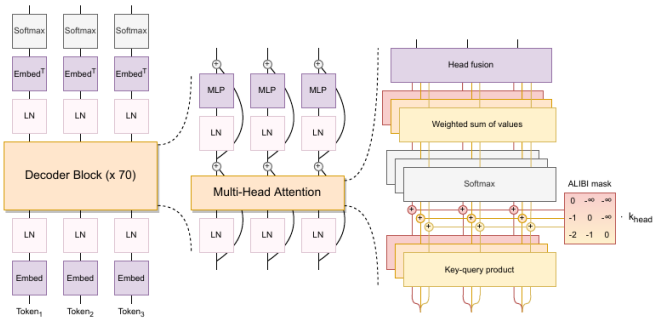

Figure 9: The BLOOM architecture example sourced from [13].

relationship that model size should be doubled for every doubling of training tokens. Over 400 language models ranging from 70 million to over 16 billion parameters on 5 to 500 billion tokens are trained to get the estimates for compute-optimal training under a given budget. The authors train a 70B model with the same compute budget as Gopher (280B) but with 4 times more data. It outperforms Gopher [116], GPT-3 [6], and others on various downstream tasks, after fine-tuning.

AlexaTM [122]: An encoder-decoder model, where encoder weights and decoder embeddings are initialized with a pre-trained encoder to speed up training. The encoder stays frozen for the initial 100k steps and is later unfrozen for end-to-end training. The model is trained on a combination of denoising and causal language modeling (CLM) objectives, concatenating a [*CLM*] token at the beginning for mode switching. During training, the CLM task is applied for 20% of the time, which improves the in-context learning performance.

PaLM [15]: A causal decoder with parallel attention and feed-forward layers similar to Eq. 4, speeding up training by a factor of 15. Additional changes to the conventional transformer model include SwiGLU activation, RoPE embeddings, multi-query attention that saves computation cost during decoding, and shared input-output embeddings. During training, loss spiking was observed, and to fix it, model training was restarted from a 100-step earlier checkpoint by skipping 200-500 batches around the spike. Moreover, the model was found to memorize around 2.4% of the training data at the 540B model scale, whereas this number was lower for smaller models.

PaLM-2 *[123]:* A smaller multi-lingual variant of PaLM, trained for larger iterations on a better quality dataset. PaLM-2 shows significant improvements over PaLM, while reducing training and inference costs due to its smaller size. To lessen toxicity and memorization, it appends special tokens with a fraction of pre-training data, which shows a reduction in generating harmful responses.

U-PaLM [124]: This method trains PaLM for 0.1% additional compute with the UL2 (also named as UL2Restore) objective [125], using the same dataset it outperforms the baseline significantly on various NLP tasks, including zero-shot, few-shot, commonsense reasoning, CoT, etc. Training with UL2R involves converting a causal decoder PaLM to a non-causal decoder PaLM and employing 50% sequential denoising, 25% regular denoising, and 25% extreme denoising loss functions.