Efficient Methods for Natural Language Processing: A Survey

Marcos Treviso[10]*, Tianchu Ji[3]*, Ji-Ung Lee[7]*, Betty van Aken[8], Qingqing Cao[2], Manuel R. Ciosici[9], Michael Hassid[1], Kenneth Heafield[13], Sara Hooker[5], Pedro H. Martins[10], André F. T. Martins[10], Peter Milder[3], Colin Raffel[6], Edwin Simpson[4], Noam Slonim[12], Niranjan Balasubramanian[3], Leon Derczynski[11], Roy Schwartz[1]

[1]The Hebrew University of Jerusalem, [2]University of Washington, [3]Stony Brook University, [4]University of Bristol, [5]Cohere For AI, [6]University of North Carolina at Chapel Hill, [7]Technical University of Darmstadt, [8]Berliner Hochschule für Technik, [9]University of Southern California, [10]IST/University of Lisbon & Instituto de Telecomunicações, [11]IT University of Copenhagen, [12]IBM Research, [13]University of Edinburgh

Abstract

Getting the most out of limited resources allows advances in natural language processing (NLP) research and practice while being conservative with resources. Those resources may be data, time, storage, or energy. Recent work in NLP has yielded interesting results from scaling; however, using only scale to improve results means that resource consumption also scales. That relationship motivates research into *efficient* methods that require less resources to achieve similar results. This survey relates and synthesises methods and findings in those efficiencies in NLP, aiming to guide new researchers in the field and inspire the development of new methods.

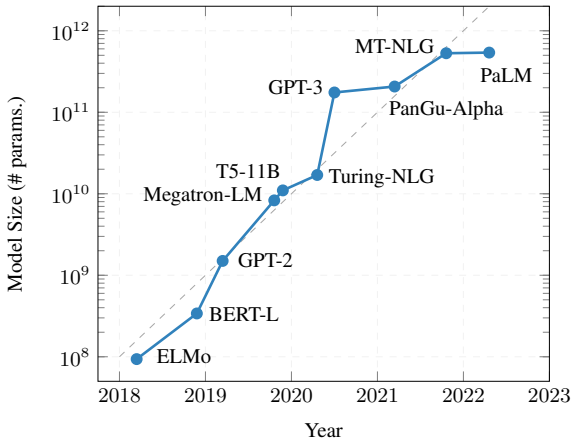

Figure 1: Evolution of the size of large pretrained language models. Adapted from Lakim et al. (2022).

1 Introduction

Training increasingly large deep learning models has become an emerging trend in the past decade (Fig. 1). While the steady increase of model parameters led to state-of-the-art performance and new research directions such as prompting, this also becomes increasingly problematic. First, such models often have restricted access, hence are not democratized, or even if so, still require a substantial amount of compute resources to run (Zhan et al., 2021). Second, they are not sustainable and require large amounts of energy for training and inference (Schwartz et al., 2020a). Third, models cannot be scaled-up indefinitely as their size is limited by the available hardware (Thompson et al., 2020). To tackle these limitations, methods that focus on improving *efficiency* are becoming increasingly popular.

Definition. Efficiency is commonly referred to as the relation between resources going into a system and its output, with an efficient system producing outputs without a waste of resources. For NLP in particular, we consider efficiency as the cost of a model in relation to the results it produces:

$$\text{Cost}(R) \propto E \cdot D \cdot H \tag{1}$$

Equation (1) describes the training cost of an AI model producing a certain (R)esult as proportional to three (non-exhaustive) factors: (1) the cost of model execution on a single (E)xample, (2) the size of the training (D)ataset and (3) the number of training runs required for model selection or (H)yperparameter tuning (Schwartz et al., 2020a). The $\text{Cost}(\cdot)$ can then be measured along multiple dimensions such as the computational, time-wise, or environmental cost. Each of them can be further quantified in multiple ways; for instance, computational cost may include the total number of floating point operations (FLOPs) or the number of model parameters. As using a single cost indicator can be misleading (Dehghani et al., 2021), this survey will collect and organize works on efficient NLP across multiple facets and discuss which dimensions can be beneficial for what use cases and stakeholders.

*Equal contribution.

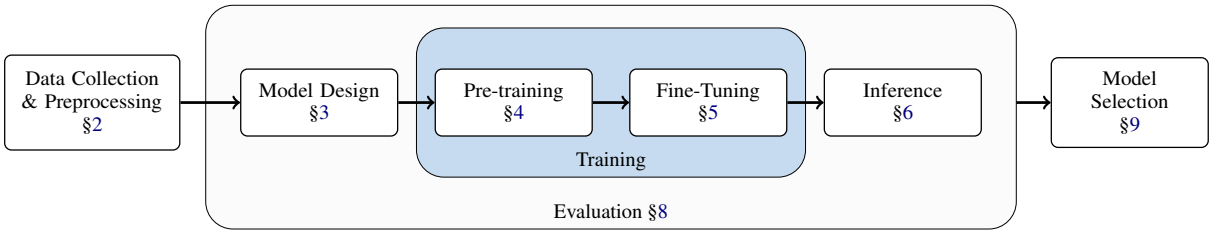

Figure 2: Schematic overview of the efficient NLP stages covered in this paper.

Scope of this survey. Our goal is to provide a gentle introduction into the broad range of methods that aim to improve efficiency with a focus on NLP. We thus structure this survey by following the typical NLP model pipeline (Fig. 2) and present the existing methods that aim to make the respective stage more efficient. To provide a practical guide to efficiency for NLP researchers, we address this work to two groups of readers: (1) Researchers from all fields of NLP working with limited resources. Depending on the bottleneck of resources, readers can directly jump to one of the covered aspects of the NLP pipeline. For instance, if the main limitation is to be expected at inference time, the methods described in Section 6 are the most relevant ones. (2) Researchers interested in improving the state-of-the-art in efficiency methods in NLP. Here, the study can serve as an entry point to find opportunities for new research directions. To guide the reader, we present a diagram with the typology of efficient NLP methods considered in this survey in Fig. 3. Moreover, while hardware choices can have a large impact on the efficiency of models, most NLP researchers do not have direct control over decisions regarding hardware, and most hardware optimizations can be employed swimmingly during all stages of the pipeline. We hence focus our work on algorithmic approaches, but provide appropriate pointers regarding hardware in Section 7. Finally, we further discuss how to quantify efficiency, what factors to consider during evaluation, and how to decide upon the best suited model.

2 Data

One way to increase efficiency can be to use less training instances and/or to better utilize the available ones. In this survey, we focus on approaches that aim to reduce the training data under the assumption that the provided labels are correct.[1]

[1]For erroneous labels we refer to Northcutt et al. (2021); Paullada et al. (2021); Kreutzer et al. (2022); Klie et al. (2022).

2.1 Filtering

Recent works show that improving *data quality* can substantially boost the performance while reducing training costs (in contrast to increasing the *data quantity*). For instance, Mishra and Sachdeva (2020) find that using ∼2% of the SNLI data (Bowman et al., 2015) can achieve comparable performances to using the full data. Lee et al. (2022b) show that removing duplicates during pre-training can already substantially increase training efficiency with equal or even better model performance. Similar trends are found in the development process of recent models such as OPT (Zhang et al., 2022) that include a deduplication step. Finally, various works focus on better understanding how individual instances contribute towards a model's performance (Swayamdipta et al., 2020). Look for

2.2 Curriculum Learning

Curriculum learning aims to increase data efficiency by finding a good ordering of the available training instances (Elman, 1993; Bengio et al., 2009). Similar trends have been observed by Dodge et al. (2020) for transformer models.

Heuristic approaches. Many approaches opt for an easy-instances-first ordering by heuristically estimating the instance difficulty. For transformer architectures, Platanios et al. (2019) find that considering the competence of the model can further improve performance and reduce training time in neural machine translation (NMT). Similar results have been observed in natural language understanding (Xu et al., 2020) and question answering (Tay et al., 2019). For language modeling, Press et al. (2021) show that an initial training on short sequences can substantially reduce training time while retaining model performance. Agrawal et al. (2021) further investigate binning training instances based on their complexity and achieve comparable performances with less training steps.

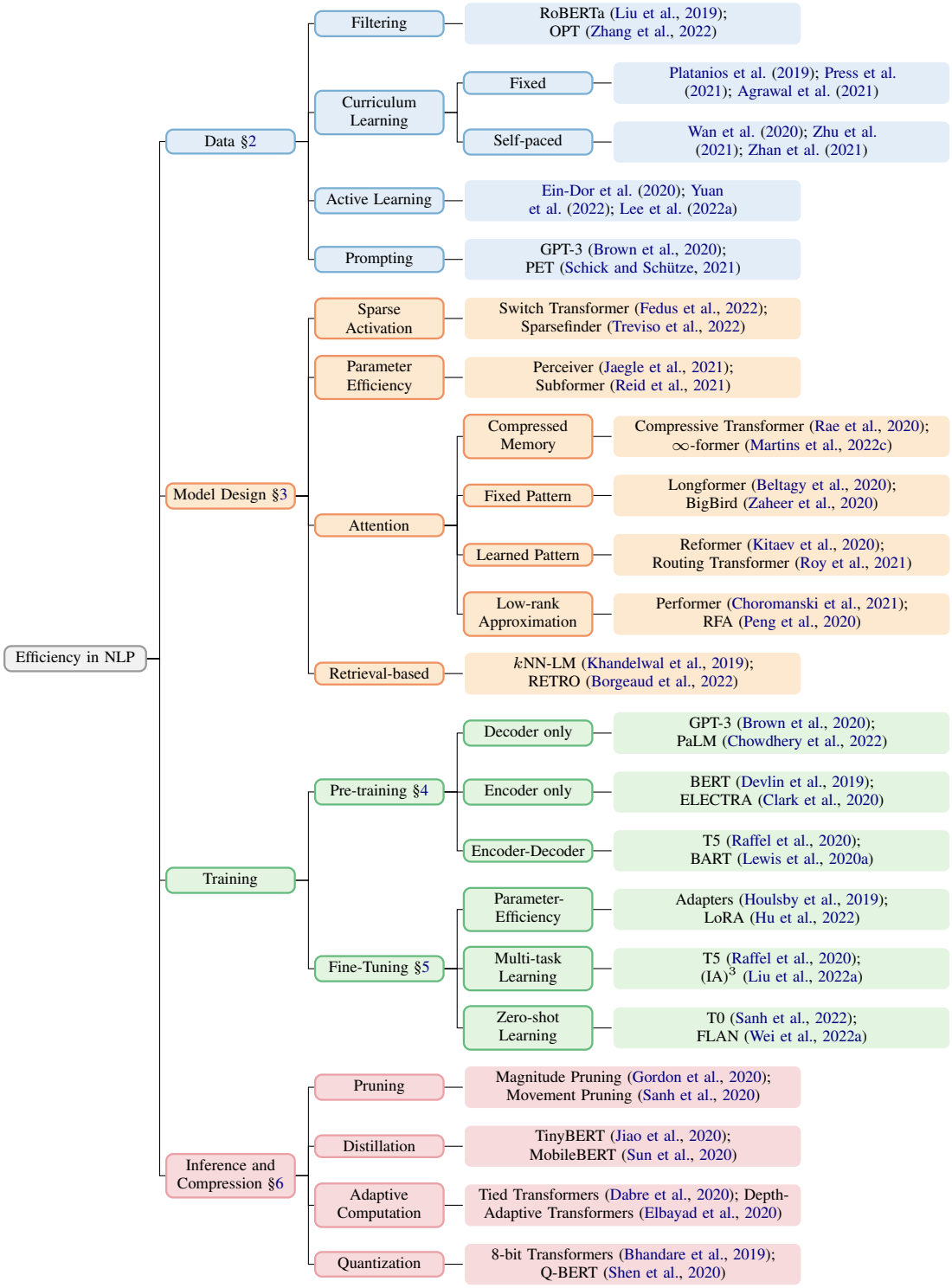

Figure 3: Typology of efficient NLP methods.

Self-paced Learning. Instead of using heuristics such as sentence length or word rarity (Platanios et al., 2019; Zhang et al., 2019; Zhao et al., 2020; Xu et al., 2020), *self-paced learning* adaptively selects instances that would be useful for model training (Kumar et al., 2010). Self-paced learning has been successfully applied in NMT using the model and data uncertainty (Wan et al., 2020; Zhou et al., 2020) and dialog generation coupled with knowledge distillation (Zhu et al., 2021). Zhan et al. (2021) even propose to learn *meta curricula* that would transfer well to other domains.

2.3 Active Learning

Data efficiency can be improved even before training by selectively annotating instances that are most helpful for model training (Settles, 2012; Ren et al., 2021b). The key challenge is to assess the helpfulness without knowing the actual label. Existing approaches thus often use a model's *uncertainty* or the underlying instance *representation* (or both) for sampling. Uncertainty-based approaches assume that instances with the highest uncertainty add the most information once labeled (Lewis and Gale, 1994) and focus on good uncertainty estimates (Tang et al., 2002; Gal et al., 2017; Yuan et al., 2020). Representation-based approaches instead focus on maximizing the diversity of selected instances (Bodó et al., 2011; Sener and Savarese, 2018; Gissin and Shalev-Shwartz, 2019; Kirsch et al., 2019). Although various works show the potential of active learning for NLP (Ein-Dor et al., 2020; Lee et al., 2020; Yuan et al., 2022), there are still open questions about its generalizability to different tasks and models (Lowell et al., 2019). Other issues are outliers in the data that can be harmful for uncertainty-based strategies (Karamcheti et al., 2021) and the potential increase in annotation difficulty and consequently, annotation cost (Settles et al., 2008; Lee et al., 2022a).

2.4 Prompting

Inspired by human interactions with models such as GPT-3 (Brown et al., 2020), prompting refers to asking the model to perform a predictive task by casting it as a textual input (Liu et al., 2021a). The final prediction is then inferred from the output of the language model (Li and Liang, 2021). In general, prompts can be either crafted manually or automatically using fill-in templates or prefix strings for token, span, and sentence-level completion (Petroni et al., 2019; Brown et al., 2020; Shin et al., 2020; Li and Liang, 2021). This makes prompting applicable to more challenging NLP tasks, such as question answering, summarization, and machine translation (Schick and Schütze, 2021). Since no training nor fine-tuning is required, prompting emerges as an efficient alternative for handling NLP tasks in an *unsupervised* fashion.[2]

3 Model Design

An active area of research is in designing more efficient models, either by implementing architectural changes or by attaching new modules that accelerate the workflow of the main model. In this section we will outline current developments made in transformers, e.g., by adapting its architecture or combining it with external resources.

3.1 Sparse Activations

As Derczynski (2020) show, the choice (and implementation) of the activation function can make an order of magnitude difference on the execution time. To accelerate inference by leveraging sparse activations, Fedus et al. (2022) propose the Switch Transformer, which routes computation to dedicated specialists ("experts"). This approach is based on a mixture of experts architecture (Jacobs et al., 1991; Shazeer et al., 2017) and can scale to up to a trillion parameters given enough memory bandwidth, which is often the bottleneck that grows with the number of experts (Rajbhandari et al., 2022). Another example of sparse activations is the adaptively sparse transformer model (Correia et al., 2019), which replaces the (dense) softmax activation in attention heads by (sparse) entmax activations, optimally learning the propensity of sparsity of each head automatically from the data. Building on this, Sparsefinder (Treviso et al., 2022) allows a more efficient attention mechanism for transformers by identifying the sparsity pattern of entmax attention before computing it.

3.2 Parameter Efficiency

Some works investigate reducing the number of parameters; for instance, by sharing weights across layers of the model, such as Universal Transformers (Dehghani et al., 2019) and ALBERT (Lan et al., 2019). Perceiver (Jaegle et al., 2021) suggests a similar approach, but inserts the original input within any inner layer. ALBERT further uses matrix decomposition to reduce the size of the embedding layer, which is one of the largest consumer of model parameters. Finally, Subformer (Reid et al., 2021) investigates ways for weight sharing in Transformers, and shows that sharing only the middle layers of the model works better than the alternatives.

3.3 Attention in Transformers

doesn't apply for sentences

A limitation of attention mechanisms in transformer models is their quadratic dependency on the sequence length, leading to variants that focus on efficient attention for long-range sequences. Existing strategies include better utilizing already

[2] See the survey of (Liu et al., 2021a) for more information.

processed segments, such as via recurrence to connect multiple segments (Transformer-XL; Dai et al. 2019), learning a network to compress a longer-term memory (Compressive Transformer; Rae et al. 2020), separately modeling global and local attention (Ainslie et al., 2020), and modeling long sequences as a continuous-time signal (∞-former; Martins et al. 2022c). Another line of research seeks to reduce the quadratic bottleneck of self-attention by using fixed attention patterns (Longformer; Beltagy et al. 2020, Sparse Transformer; Child et al. 2019, BigBird; Zaheer et al. 2020), or learning attention sparsity patterns by grouping tokens into buckets or clusters (Reformer; Kitaev et al. 2020, SMYRF; Daras et al. 2020, Routing Transformer; Roy et al. 2021). Some strategies modify the attention mechanism by deriving low-rank approximations to the query-key matrices via a reverse application of the kernel trick that renders linear runtime, as in Linear Transformer (Katharopoulos et al., 2020), Performer (Choromanski et al., 2021), and RFA (Peng et al., 2020).[3] Finally, S4 (Gu et al., 2022) is a recent alternative to transformers that leverages a discretization of state space representations and a parameterization of the state matrix, and achieves strong results for very long inputs.

3.4 Retrieval-Augmented Models

A promising direction in text generation is to combine parametric models with retrieval mechanisms, leading to semi-parametric models (Gu et al., 2018; Lewis et al., 2020b).[4] At inference time, the model retrieves tokens / phrases / sentences from a database, which are then used by the model through interpolation of probability distributions (Khandelwal et al., 2019), gating mechanisms (Yogatama et al., 2021), or attention (Borgeaud et al., 2022). This typically amounts to trading model size with the number of database entries. E.g., RETRO (Borgeaud et al., 2022) matches the performance of GPT-3, Jurassic-1 (Lieber et al., 2021), and Gopher (Rae et al., 2021) despite having 25 times fewer parameters, by retrieving chunks of tokens from a 2 trillion token database.

These models also have good generalization properties: by retrieving from domain-specific databases, models can be applied to domains not seen during training (Khandelwal et al., 2019, 2021), avoiding the need to fine-tune the model for each domain. Having an explicit memory also allows retrieval-augmented models to be adapted "on-the-fly". For instance, Martins et al. (2022b) show that adding corrected examples to a database leads to better translations than fine-tuning while reducing the total translation time. A downside, however, is that retrieval-augmented models are generally slow, since they need to perform retrieval during inference. Several recent works proposed strategies to alleviate this issue, such as pruning the database, having smaller input-dependent databases, reducing the representations dimension, caching information, and reducing the number of retrieval steps (He et al., 2021a; Meng et al., 2022; Wang et al., 2021b; Martins et al., 2022a,b; Alon et al., 2022).

4 Pre-training

Pre-training is a common step in developing NLP models (Peters et al., 2018; Devlin et al., 2019). It typically involves a form of self-supervision of large amounts on textual data, such as prediction of masked words (e.g., BERT) or language modeling (e.g., GPT family of models). The pre-trained models are subsequently fine-tuned for specific tasks (Section 5). In addition to improving performance, the pre-training step can significantly improve efficiency (Peters et al., 2018; Kovaleva et al., 2019). For example, He et al. (2019); Neyshabur et al. (2020) show that pre-training improves convergence speed on downstream tasks. As Fig. 1 shows, the increase in size of these models has been constant for the past several years and has revealed capabilities that only emerge once models become very large (Wei et al., 2022b). However, pre-training these increasingly large models is computationally demanding (Strubell et al., 2019; Schwartz et al., 2020a), leading to the important challenge of reducing their costs.

4.1 Dynamic Masking

The choice of the objective task can determine the success of the pre-trained model when applied on downstream tasks. Self-supervised learning objectives have been a key component for pre-training models on large amounts of unlabeled data. These objectives vary depending on whether the task is modeled using a decoder, an encoder, or both.

Decoder only. The classic objective function for decoder only models, such as GPT (Radford et al.,

[3] See Tay et al. (2020) for a survey on efficient attention.

[4] See the survey by Li et al. (2022) for a comprehensive overview of retrieval-augmented text generation models.

2019; Brown et al., 2020) and PaLM (Chowdhery et al., 2022), is the *causal language modeling* (CLM) objective, which predicts the next word given a prefix using the cross-entropy loss over the whole vocabulary.

Encoder only. A common way for pre-training encoder only models is presented by BERT (Devlin et al., 2019), which uses two objective tasks: (1) The *masked language model* (MLM) task, aiming at filling randomly masked tokens of a textual input, and (2) the *next sentence prediction* (NSP) task, with the goal of predicting whether the two random sentences appear consecutively in the training data. To make better use of the available data, various works have investigated masking strategies that differ from the static masking used in BERT. For instance, Liu et al. (2019) show that dynamically masking tokens during training—i.e., randomly masking 15% of the tokens at each step instead of masking them once before training—can already improve efficiency with a comparable performance. In addition, they show that the NSP objective can be dropped from the pre-training phase in order to get better model performance.

Other works show that masking specific tokens (such as objects or content words; Bitton et al., 2021) or more tokens (Wettig et al., 2022) leads to higher performance and more efficient use of the available data. ELECTRA (Clark et al., 2020) and DeBERTa (He et al., 2021b) experiment with *replaced token detection* (RTD), a new self-supervised learning objective that uses a small generator model to replace tokens in the input. Both works show that RTD leads to faster and better performing pre-training compared to BERT.

Encoder-Decoder Another approach, suggested in T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a), uses a denoising *sequence-to-sequence* objective to pretrain an encoder-decoder LM, allowing the decoder to predict a span of tokens for masked positions rather than a single token.

5 Fine-Tuning

Fine-tuning refers to the step of adapting a pre-trained model to a new downstream task. In general, fine-tuning specifically refers to gradient-based training on downstream task data. In this survey, we use a broader definition of fine-tuning that includes any method used to apply a pre-trained model to a downstream task.

5.1 Parameter-Efficient Fine-Tuning

Gradient-based fine-tuning typically involves training all of a model's parameters on downstream task data. This means that each time a pre-trained model is fine-tuned on a new task, an entirely new set of model parameters is created. If a model is fine-tuned on many tasks, the storage requirements can become onerous. The seminal ELMo work originally adapted a pre-trained model to downstream tasks by training a new classification layer and leaving the rest of the parameters fixed. This approach updates dramatically fewer parameters than training the full model but has been shown to produce worse performance and has therefore become less common (Devlin et al., 2019).

An alternative is *parameter-efficient fine-tuning* (PEFT), which aims to adapt a model to a new task while only updating or adding a relatively small number of parameters. Adapters (Houlsby et al., 2019; Bapna and Firat, 2019; Rebuffi et al., 2017), which inject new trainable dense layers into a pre-trained model, were the first PEFT method proposed for NLP models. Adapters have recently been improved by the "Compacter" method of (Karimi Mahabadi et al., 2021), which constructs the adapter parameter matrices through Kronecker products of low-rank matrices. As an alternative to adding new layers, parameter-efficiency can be achieved by directly modifying activations with learned vectors, either by concatenation (Lester et al., 2021; Li and Liang, 2021), multiplication (Liu et al., 2022a), or addition (Ben Zaken et al., 2022). Alternatively, rather than adding new parameters or changing the model's computational graph, it is possible to make updates to the original model cheaper to store through the use of sparse (Sung et al., 2021; Guo et al., 2021) or low-rank (Hu et al., 2022) updates. Finally, it has been shown that optimization can be performed in a low-dimensional subspace (Li et al., 2018); storing the updates in this subspace can be seen as a PEFT method (Aghajanyan et al., 2021b). State-of-the-art PEFT methods add or update roughly four orders of magnitude fewer parameters than full-model fine-tuning without sacrificing (and in some cases improving) performance (Hu et al., 2022; Karimi Mahabadi et al., 2021; Liu et al., 2022a).

5.2 Multi-Task and Zero-Shot Learning

While traditional transfer learning includes fine-tuning, there are other paradigms that allow for

immediate application of a pre-trained model to a downstream task of interest. *Multi-task learning* (Caruana, 1997; Ruder, 2017) aims to train a single model that can perform a wide variety of tasks out of the box. Typically, this is done by explicitly training the model on data from all tasks of interest. If a multi-task model has already been trained on a given downstream task, then no fine-tuning is necessary. Recent work has additionally demonstrated that multi-task models are also amenable to fine-tuning (Raffel et al., 2020; Aghajanyan et al., 2021a; Aribandi et al., 2022; Liu et al., 2022a).

In certain cases, a multi-task model can be applied to a new task without any fine-tuning. This ability is referred to as *zero-shot generalization*. Radford et al. (2017, 2019) and Brown et al. (2020) demonstrated that language models trained with an unsupervised objective were able to perform a variety of tasks out-of-the-box. Later, Sanh et al. (2022) and Wei et al. (2022a) showed that multitask training can also enable zero-shot generalization abilities. While zero-shot generalization can circumvent fine-tuning completely, it has (as of writing) only been demonstrated on large and computationally-intensive models.

6 Inference and Compression

Various approaches have been proposed to improve efficiency at inference time. *Compression* methods such as *pruning* (LeCun et al., 1989) and *distillation* (Hinton et al., 2015) assume that smaller models are more efficient than larger models. *Adaptive computation* works accelerate inference by ignoring inner modules for making a prediction (Schwartz et al., 2020b). Finally, *quantization* is an orthogonal approach that directly increases efficiency by modifying the underlying data type.

6.1 Pruning

Initially proposed by LeCun et al. (1989), removing unnecessary weights from a neural network aims to avoid unnecessary computation to reduce inference time with limited accuracy loss, and furthermore, decrease memory capacity and bandwidth requirements. Pruning can be applied on different levels within a model: for instance, Voita et al. (2019); Michel et al. (2019) find that only few attention heads substantially contribute towards a model's prediction and propose to prune the rest; Correia et al. (2019); Ji et al. (2021); Qu et al. (2022) verified that the weak attention values in the transformers can be pruned without accuracy loss. Others focus on pruning individual weights (Sanh et al., 2019; Gordon et al., 2020) or layers (Dong et al., 2017; Sajjad et al., 2020). Finally, some works try to identify good criteria for pruning specific weights/layers (Sanh et al., 2020; Hoefler et al., 2021) or even propose to dynamically drop layers (Fan et al., 2020) during inference; sometimes in combination with other efficiency methods such as adapters (Rücklé et al., 2021). The increasing popularity of pruning methods has further raised the question of how to quantify and compare them (Tessera et al., 2021; Blalock et al., 2020; Gale et al., 2019).[5]

6.2 Distillation

Whereas pruning primarily focuses on removing weights from a pre-trained model, Hinton et al. (2015) instead propose to train a smaller model (student) from scratch by using the pre-trained model to obtain a supervision signal (teacher). While early works focus on distilling task-specific models (Kim and Rush, 2016), recent works focus on distilling pre-trained models that can then be fine-tuned on specific downstream tasks (Sanh et al., 2019; Liu et al., 2020; Jiao et al., 2020; Sun et al., 2020).

6.3 Adaptive Computation

An alternative to compression approaches can be to adaptively decide for each instance which part of a model to use. For example, *early exit predictions* allow a system to only utilize the outputs of lower (early) layers in a model to make a prediction (Dabre et al., 2020; Elbayad et al., 2020; Schwartz et al., 2020b; Xin et al., 2020).

6.4 Quantization

Various data types can be utilized as the underlying representation in neural networks (Section 7). Mapping high-precision data types to low-precision ones is commonly referred to as *quantization*. While quantization saves memory and computational cost, reducing the precision can lead to a loss in terms of accuracy. Therefore, quantization often requires a careful model construction and training.

Low- and mixed-precision. Various works target specific precision-levels such as integers (Kim et al., 2021), 8-bit (Quinn and Ballesteros, 2018; Zafrir et al., 2019; Bhandare et al., 2019; Prato

[5] A detailed taxonomy of different pruning approaches is introduced by Hoefler et al. (2021).

et al., 2020) and 3-bit quantization (Ji et al., 2021; Zadeh et al., 2022), and even ternary and binary representations (Zhang et al., 2020; Bai et al., 2020). Other works investigate mixed-precision quantization as different components may have a different sensitivity regarding their underlying precision. For instance, Shen et al. (2020) show that embedding layers require more precise parameter representations than the attention layer while Kim et al. (2021) show that nonlinear functions require more bits than the general matrix multiplication. Others define quantization as a constrained optimization problem to automatically identify layers where a lower precision is sufficient (Hubara et al., 2021). These works show that customized quantization schemes across different components can maintain the accuracy while increasing efficiency.

Quantization-aware training. Finally, several works propose to consider quantization already during training to make them robust against performance losses after quantization (Zafrir et al., 2019; Kim et al., 2021; Stock et al., 2021). For instance, Bai et al. (2020); Zhang et al. (2020) propose to utilize knowledge distillation to maintain the accuracy of binarized and ternarized models.

6.5 Other Methods

Although this survey presents the most prominent research areas that aim to improve inference efficiency, there exist several other methods with the same goal. For instance, Wu et al. (2022) combine several methods to achieve utmost model compression, while other works improve task-specific mechanisms, such as beam-search in machine translation (Peters and Martins, 2021). Moreover, parallelism can also be exploited to further increase inference efficiency (Rajbhandari et al., 2022).

7 Hardware Utilization

Finally, we discuss several methods that consider the underlying hardware used for training and inference. A majority of the effort is dedicated to reducing GPU memory consumption as one of the major bottlenecks in transformer models. Note that many of the presented techniques can be applied across different stages of training and inference (Fig. 2), and can be combined for further efficiency.

Data types. Traditionally, neural networks use the IEEE 765 single-precision 32-bit float which consists of 4 bytes representing a floating point number (`float32`). However, this substantially affects the memory consumption in GPUs with the increase of model parameters. Combining half-precision (`float16`) and single-precision (`float32`) data representations can cut a network's memory consumption in half for inference and almost half for training (Micikevicius et al., 2018). An alternative to `float16` is the Brain Floating Point (`bfloat16`) that is utilized in TPUs and can lead to a more stable training (Kalamkar et al., 2019). `bfloat16` and `float16` can furthermore lead to double the FLOP/S due to hardware support in many modern CPUs and GPUs.

Reducing optimizer memory. Because the Adam optimizer keeps track of first and second order momentum, it needs to store two floats for each parameter in the neural network. Therefore, to train a model containing K parameters, the GPU must store $3K$ parameters corresponding to the model, the first and the second order momentum. Libraries like DeepSpeed (Ren et al., 2021a) allows the optimizer to be offloaded from GPU memory and into CPU RAM where the computations are performed in the CPU using highly-efficient AVX instructions. `bitsandbytes` (Dettmers et al., 2022) uses dynamic block-wise quantization and 8-bit integers to represent optimizers. Block-wise quantization requires `bitsandbytes` to split each tensor into blocks that are individually quantized, reducing the inter-GPU communication. `bitsandbytes` can reduce Adam's GPU memory consumption by 75% and, in many cases, speed up training by reducing inter-GPU communication. While `bitsandbytes` runs on GPUs, the method is theoretically compatible with the optimizer offloading presented in DeepSpeed.

Specialized hardware. Specialized hardware for NLP applications that utilizes Application Specific Integrated Circuits (ASICs) or Field Programmable Gate Arrays (FPGAs) exists, but is not broadly available. These hardware designs use dedicated computational units for irregular methods that improve efficiency (such as quantization and pruning discussed in Section 6), hence they improve the efficiency. For example, Zadeh et al. (2020, 2022); Li et al. (2021); Qu et al. (2022) managed to support ultra-low-bit and mixed precision computation that cannot be done on CPUs/GPUs; Ham et al. (2020, 2021); Qu et al. (2022); Wang et al. (2021a) proposed hardware that predicts the

unnecessary components in the transformers and prunes them, including redundant heads/tokens and weak attention values. Qu et al. (2022) specifically proposed specialized hardware to schedule data loading to alleviate the imbalance introduced by pruning. Other works develop new types of dedicated processors and memories to match the properties of the components in the transformers; for instance, softmax and layer normalization (Lu et al., 2020; Liu et al., 2021b), and embedded Resistive RAM (a nonvolatile memory with low latency and energy comsumption) to store word embeddings (Tambe et al., 2021).

Co-design. Finally, we provide some pointers for works that jointly optimize the design of hardware, software, and algorithms which historically has been an important driver of efficiency gains (Hooker, 2021). For instance, Lepikhin et al. (2021) demonstrate that improving the underlying compiler can already substantially improve parallelization allowing them to scale their model up to 600B parameters. Other prominent examples for co-design also focus on improvements of mixture of experts models that consider the underlying hardware leading to substantial speedups (He et al., 2022; Rajbhandari et al., 2022). Lastly, Barham et al. (2022) propose a novel gang-scheduling approach together with parallel asynchronous dispatch that further leads to substantial efficiency improvements.

7.1 Edge Devices

Running advanced NLP models to resource-constrained devices provides better user experience as it preserves user-privacy and reduces inference latency. Various works specifically target increasing the efficiency for on-device settings. SqueezeBERT (Iandola et al., 2020) is a mobile BERT-like architecture that incorporates efficient group convolutions into self-attention and it runs faster on mobile devices than other efficient models like MobileBERT (Sun et al., 2020). EdgeFormer (Ge and Wei, 2022) is a lightweight encoder-decoder transformer architecture that is designed for on-device settings. It runs on mobile CPUs under low latency and provides high-quality machine translation and grammar error correction abilities. GhostBERT (Huang et al., 2021) uses ghost modules that are built on top of depthwise separable convolutions used in MobileNets (Howard et al., 2017). LiteTransformer (Wu* et al., 2020) utilizes long-short range attention to encode local context by convolutions and captures long range dependencies by attention operations. It improves the Transformer performance on machine translation tasks by a large margin under resource-constrained settings. Finally, ProFormer (Sankar et al., 2021) uses locality sensitive hashing and local projection attention layers to build word embeddings for text classification and reduces the runtime and memory for on-device deployments.

8 Evaluation

To evaluate efficiency, it is important to establish what resource—e.g., money, data, memory, time, power consumption, carbon emissions, etc—one attempts to constrain. Furthermore, efficiency does not intrinsically guarantee a reduction in overall resource consumption, as the resulting cost reduction may lead to an increase in demand and counteract its gains. This is known as Jevons paradox (Jevons, 1866) and is in part moderated by time lag between efficiency gains and demand increase, and external, human-influenced factors such as energy pricing and regulation.

8.1 Measuring Efficiency

There are often multiple factors that need to be traded-off against each other when improving efficiency. For instance, while a longer training of models may increase their task performance, at the same time, it increases the resource consumption.

Pareto optimality. One solution for this issue can be to identify Pareto-optimal solutions, those for which no other system reaches a better or equal task performance with lower resource consumption. As there still may be more than one Pareto-optimal solution, the final choice depends on the application context; e.g., a small, average-quality model and a large, high-quality model can both be optimal. Consequently, as long as a model contributes to or extends the Pareto-optimal curve for a given problem and measurement space, it contributes something new—even if other solutions may use less resources or produce higher quality scores. Advancing NLP through pushing Pareto barriers is an established practice. For instance, the WNGT 2020 machine translation shared task (Birch et al., 2020) considers the Pareto frontier between real time taken, system or GPU memory usage, and model size, as well as BLEU. Especially in MT evaluation, such trade-offs are commonplace (Kim

et al., 2019; Bogoychev et al., 2020; Behnke and Heafield, 2021). Puvis de Chavannes et al. (2021) include power consumption as a trade-off against perplexity to explore Pareto-efficient hyperparameter combinations for transformer models. Finally, Liu et al. (2022b) examine Pareto efficiency for a number of tasks in an attempt to narrow model selection search space to efficient examples.

Power consumption. One resource to measure efficiency is power consumption. There exist various way to measure power consumption, for instance, by using specific hardware such as an electricity meter. While this can provide precise figures with a high temporal accuracy, it cannot provide a fine-grained estimate. Moreover, this does not cover external energy costs such as cooling or networking. Another way is to utilize software tools such as MLCO2 (Luccioni et al., 2019). Some tools even provide a real-time breakdown of the power consumption of different components within a machine (Henderson et al., 2020) or local machine API-reported figures to stop training early if prudent (Anthony et al., 2020). Finally, Hershcovich et al. (2022) introduce a model card for NLP systems that encourages researchers to document efficiency in a consistent manner. Note that measuring power consumption programmatically comes with a number of caveats. First, sampling frequency is often restricted for a number of reasons at various levels of the stack and may result in a lag in measurement start. Consequently, shorter experiments may log an energy use of zero, and there will almost always be some part of a process' real energy demand that is missed. Second, inefficiencies such as heat loss are not reported by current APIs and hence, does not cover cooling and other system management activities. Third, not all architectures and operating systems are supported. For instance, power consumption under OSX is difficult to manage, and direct figures for TPU power consumption are not available.

Carbon emission. Besides power consumption, the aforementioned works often also report the carbon emissions. They are computed using the power consumption and the carbon intensity of the marginal energy generation that is used to run the program. Thus, low-energy does not mean low-carbon, and high-energy models can—in the right region and with some care—be zero-carbon in terms of point energy consumption impact, if executed at the right time (i.e., when the energy mix is low-carbon intensity). For estimating the CO2 emissions from a specific program execution, APIs such as ElectricityMap[6] provide real-time access to carbon intensity for many regions. However, as carbon intensity varies and is affected by other factors like the power usage efficiency in a data center, it is often a poor basis for comparison; in fact, Henderson et al. (2020) recommend to use multiple runs for a stable estimate. Furthermore, one needs to consider that zero-carbon program executions still consume energy (to beware of Jevons paradox).

Financial impact. Monetary cost is a resource that one typically prefers to be efficient with. Both fixed and running costs affect NLP, depending on how one chooses to execute a model. As hardware configurations and their prices form discrete points on a typically non-linear scale, it is worth paying attention to efficient cost points and fitting to these. Implementing pre-emptible processes that can recover from interruptions also often allows access to much cheaper resources. When calculating or amortizing hardware costs, one should also factor in downtime, maintenance, and configuration. Measuring the total cost of ownership (TCO) provides a more useful metric.

FLOP/s. Finally, a frequently reported efficiency measure are the floating point operations (FLOPs) and floating points per second (FLOP/s). While this discrete metric sounds well-defined in terms of what the hardware does, there is some variation at multiple stages of the stack, adding uncertainty. For example, different operations may count as a FLOP on different hardware; non-floating-point operations are not considered; hardware is rarely 100% utilised and achieving this productively is a challenge, so theoretical FLOP/s performance cannot be multiplied with time elapsed to yield the amount of computing performed. Still, FLOP/s per unit power can indicate which hardware choices have the potential to offer Pareto-efficient trade-offs for these factors (Hsu et al., 2005).

8.2 Trade-offs with other Desiderata

One major, but seldomly studied concern when improving the efficiency are trade-offs with other desiderata such as fairness and robustness. For instance, Hooker et al. (2020); Renduchintala et al. (2021); Silva et al. (2021) find that compression

[6] https://electricitymap.org

techniques such as pruning can amplify existing biases; Mohammadshahi et al. (2022) further showcase this in a multilingual setting. So far, not many works investigate preserving a model's fairness when increasing its efficiency. To quantify such effects, Xu et al. (2021) propose loyalty as a novel metric. Finally, Xu and Hu (2022) attempt to study these effects more systematically, however, with mixed conclusions. While, more positive insights have been found with other desiderata such as the out-of-distribution (OOD) generalization (Ahia et al., 2021; Iofinova et al., 2022) and model transfer (Gordon et al., 2020), we find that more work is necessary to better understand the impact of efficiency methods on them.

8.3 Open Challenges in Measuring Efficiency

The choice of hardware can lead to pronounced differences in certain efficiency measurements such as latency and thoroughput (Lee-Thorp et al., 2022). Properly measuring efficiency still remains a big challenge. For instance, Cao et al. (2020) show that using software-based tools can often introduce large errors in estimating the true energy. Instead, more accurate estimates may be obtained by training a classifier on ground-truth energies obtained from power monitors (Cao et al., 2021); however, scaling them to new hardware devices and multiple GPUs stull remains an open problem.

Separating different stages. It is important to separately characterize the efficiency of pretraining and fine-tuning stages. For example, models may present different memory requirements during training yet result in trained models with comparable inference memory consumption. This is because training often involves design choice that increases the memory overhead of backward propagation. For example, certain optimizers can require significantly more memory. In a similar vein, parameter sharing techniques have few benefits during training but show memory improvements at inference (Dehghani et al., 2021).

Disagreement between cost factors. As partially discussed in Section 7, cost indicators may disagree with each other. For instance, mixture of experts increase the overall parameter count, but improve the trade-off between quality and FLOPs, as they minimize the per-data cost by routing to subsections of the model (Rajbhandari et al., 2022). Conversely, unstructured sparsity techniques can significantly minimize the overall number of FLOPs. Yet in practice, it introduces low-level operations that can lead to far higher memory requirements to store the indices that indicate what part of the matrix is sparse (Qu et al., 2022). Finally, Dao et al. (2021) find specific sparsity patterns that achieve more predictable speedups with current hardware.

9 Model Selection

Hyperparameter search. The performance of machine learning methods can be substantially improved by a careful choice of hyperparameters. Model-based techniques such as Bayesian optimization (BO) (Snoek et al., 2012; Feurer et al., 2015) and graph-based semi-supervised learning (Zhang and Duh, 2020) use surrogate models to search efficiently for optimal hyperparameters, avoiding expensive grid search or manual tuning. The SMAC3 library (Lindauer et al., 2022) implements several BO strategies, including a budget-limited variant for expensive deep learning tasks, and is integrated into *auto-sklearn* (Feurer et al., 2020) and *auto-pytorch* (Zimmer et al., 2021).

A complementary approach to reduce the cost of hyperparameter optimization is the successive halving algorithm (SHA) (Jamieson and Talwalkar, 2016) and its massively parallel variant, asynchronous SHA (ASHA, Li et al. 2020), which test multiple hyperparameter settings in parallel for a fixed number of training iterations, then discard the half of the settings with the worst validation set performance. However, with limited computational budgets, both BO and ASHA can sometimes fail to identify good settings (Liu and Wang, 2021). It is unclear whether these methods can also be used to choose random initial weights or to order training samples, which also have a substantial effect on model performance (Dodge et al., 2020).

Hyperparameter transfer. To minimise the number of trials needed to find optimal hyperparameter settings, we can transfer knowledge from other datasets or tasks – similar to how an ML engineer might select reasonable settings by hand. Transfer neural processes (Wei et al., 2021) provide a way to transfer observations, parameters and configurations from previous tasks using Bayesian optimization with a neural process as the surrogate model. This can lead to more accurate models with fewer trials than conventional BO approaches, but has yet to be tested for large NLP models. Furthermore, when training a large neural network,

the cost of each tuning step can be reduced using μTransfer (Yang et al., 2021) to tune a small model, then transfer the hyperparameters to a larger model. First, the target model is parameterized using Maximal Update Parametrization (μP) (Yang and Littwin, 2021), which finds a suitable smaller model (reduced width or depth) whose optimal hyperparameters will be similar to those of the larger target model. The small model is tuned using any preferred approach, and the chosen hyperparameter values are then used directly for the large target model. μTransfer is applicable to many different hyperparameters, including learning rate, momentum, weight initialization variance and weight multipliers, but not to those controlling regularization, such as dropout.

10 Conclusion

In this survey, we organized the existing literature according to the traditional NLP pipeline, and provided a broad overview of existing methods to increase efficiency and their shortcomings. As our discussion shows, efficiency in NLP can be achieved in many different ways; but is also subject to various open challenges such as a good metric to quantify it.

Acknowledgements

This work was initiated at and benefitted substantially from the Dagstuhl Seminar 22232: *Efficient and Equitable Natural Language Processing in the Age of Deep Learning*. We further thank Yuki Arase, Jesse Dodge, Jessica Forde, Jonathan Frankle, Iryna Gurevych, Alexander Koller, Alexander Löser, Alexandra Sasha Luccioni, Haritz Puerto, Nils Reimers, Leonardo Riberio, Anna Rogers, Andreas Rücklé, Noah A. Smith, Emma Strubell, and Thomas Wolf for a fruitful discussion and helpful feedback at the seminar.

References

Armen Aghajanyan, Anchit Gupta, Akshat Shrivastava, Xilun Chen, Luke Zettlemoyer, and Sonal Gupta. 2021a. Muppet: Massive multi-task representations with pre-finetuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5799–5811, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Armen Aghajanyan, Sonal Gupta, and Luke Zettlemoyer. 2021b. Intrinsic dimensionality explains the effectiveness of language model fine-tuning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7319–7328, Online. Association for Computational Linguistics.

Ameeta Agrawal, Suresh Singh, Lauren Schneider, and Michael Samuels. 2021. On the role of corpus ordering in language modeling. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 142–154, Virtual. Association for Computational Linguistics.

Orevaoghene Ahia, Julia Kreutzer, and Sara Hooker. 2021. The low-resource double bind: An empirical study of pruning for low-resource machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3316–3333, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Joshua Ainslie, Santiago Ontanon, Chris Alberti, Vaclav Cvicek, Zachary Fisher, Philip Pham, Anirudh Ravula, Sumit Sanghai, Qifan Wang, and Li Yang. 2020. ETC: Encoding long and structured inputs in transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 268–284, Online. Association for Computational Linguistics.

Uri Alon, Frank F Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. 2022. Neuro-Symbolic Language Modeling with Automaton-augmented Retrieval. In *Procceedings of International Conference on Machine Learning*.

Lasse F Wolff Anthony, Benjamin Kanding, and Raghavendra Selvan. 2020. CarbonTracker: Tracking and predicting the carbon footprint of training deep learning models. In *Proceedings of the workshop on Challenges in Deploying and monitoring Machine Learning Systems, ICML*.

Vamsi Aribandi, Yi Tay, Tal Schuster, Jinfeng Rao, Huaixiu Steven Zheng, Sanket Vaibhav Mehta, Honglei Zhuang, Vinh Q. Tran, Dara Bahri, Jianmo Ni, Jai Gupta, Kai Hui, Sebastian Ruder, and Donald Metzler. 2022. Ext5: Towards extreme multi-task scaling for transfer learning. In *International Conference on Learning Representations*.

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jing Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2020. BinaryBERT: Pushing the Limit of BERT Quantization. *arXiv:2012.15701 [cs]*. ArXiv: 2012.15701.

Ankur Bapna and Orhan Firat. 2019. Simple, scalable adaptation for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the*

9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 1538–1548, Hong Kong, China. Association for Computational Linguistics.

Paul Barham, Aakanksha Chowdhery, Jeff Dean, Sanjay Ghemawat, Steven Hand, Dan Hurt, Michael Isard, Hyeontaek Lim, Ruoming Pang, Sudip Roy, Brennan Saeta, Parker Schuh, Ryan Sepassi, Laurent El Shafey, Chandramohan A. Thekkath, and Yonghui Wu. 2022. Pathways: Asynchronous distributed dataflow for ml.

Maximiliana Behnke and Kenneth Heafield. 2021. Pruning neural machine translation for speed using group lasso. In *Proceedings of the Sixth Conference on Machine Translation*, pages 1074–1086, Online. Association for Computational Linguistics.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The Long-Document Transformer. *arXiv:2004.05150 [cs]*. ArXiv: 2004.05150.

Elad Ben Zaken, Yoav Goldberg, and Shauli Ravfogel. 2022. BitFit: Simple parameter-efficient fine-tuning for transformer-based masked language-models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Dublin, Ireland. Association for Computational Linguistics.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 41–48.

Aishwarya Bhandare, Vamsi Sripathi, Deepthi Karkada, Vivek Menon, Sun Choi, Kushal Datta, and Vikram Saletore. 2019. Efficient 8-Bit Quantization of Transformer Neural Machine Language Translation Model. *arXiv:1906.00532 [cs]*. ArXiv: 1906.00532.

Alexandra Birch, Andrew Finch, Hiroaki Hayashi, Kenneth Heafield, Marcin Junczys-Dowmunt, Ioannis Konstas, Xian Li, Graham Neubig, and Yusuke Oda, editors. 2020. *Proceedings of the Fourth Workshop on Neural Generation and Translation*. Association for Computational Linguistics, Online.

Yonatan Bitton, Michael Elhadad, Gabriel Stanovsky, and Roy Schwartz. 2021. Data efficient masked language modeling for vision and language. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3013–3028, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. 2020. What is the state of neural network pruning? *Proceedings of machine learning and systems*, 2:129–146.

Zalán Bodó, Zsolt Minier, and Lehel Csató. 2011. Active learning with clustering. In *Active Learning and Experimental Design workshop In conjunction with AISTATS 2010*, pages 127–139. JMLR Workshop and Conference Proceedings.

Nikolay Bogoychev, Roman Grundkiewicz, Alham Fikri Aji, Maximiliana Behnke, Kenneth Heafield, Sidharth Kashyap, Emmanouil-Ioannis Farsarakis, and Mateusz Chudyk. 2020. Edinburgh's submissions to the 2020 machine translation efficiency task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 218–224, Online. Association for Computational Linguistics.

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego De Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack Rae, Erich Elsen, and Laurent Sifre. 2022. Improving language models by retrieving from trillions of tokens. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 2206–2240. PMLR.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Qingqing Cao, Aruna Balasubramanian, and Niranjan Balasubramanian. 2020. Towards accurate and reliable energy measurement of NLP models. In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 141–148, Online. Association for Computational Linguistics.

Qingqing Cao, Yash Kumar Lal, Harsh Trivedi, Aruna Balasubramanian, and Niranjan Balasubramanian. 2021. IrEne: Interpretable energy prediction for transformers. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2145–2157, Online. Association for Computational Linguistics.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating Long Sequences with Sparse Transformers. *arXiv:1904.10509 [cs, stat]*. ArXiv: 1904.10509 version: 1.

Krzysztof Marcin Choromanski, Valerii Likhosherstov, David Dohan, Xingyou Song, Andreea Gane, Tamas Sarlos, Peter Hawkins, Jared Quincy Davis, Afroz Mohiuddin, Lukasz Kaiser, David Benjamin Belanger, Lucy J Colwell, and Adrian Weller. 2021. Rethinking attention with performers. In *International Conference on Learning Representations*.

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. arXiv:2204.02311.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. Electra: Pre-training text encoders as discriminators rather than generators. In *Proc. of ICLR*.

Gonçalo M. Correia, Vlad Niculae, and André F. T. Martins. 2019. Adaptively Sparse Transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2174–2184, Hong Kong, China. Association for Computational Linguistics.

Raj Dabre, Raphael Rubino, and Atsushi Fujita. 2020. Balancing cost and benefit with tied-multi transformers. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 24–34, Online. Association for Computational Linguistics.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy. Association for Computational Linguistics.

Tri Dao, Beidi Chen, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Ré. 2021. Pixelated butterfly: Simple and efficient sparse training for neural network models.

Giannis Daras, Nikita Kitaev, Augustus Odena, and Alexandros G Dimakis. 2020. Smyrf - efficient attention using asymmetric clustering. In *Advances in Neural Information Processing Systems*, volume 33, pages 6476–6489. Curran Associates, Inc.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. 2021. The efficiency misnomer. *CoRR*, abs/2110.12894.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. 2019. Universal transformers. In *International Conference on Learning Representations*.

Leon Derczynski. 2020. Power consumption variation over activation functions. *arXiv preprint arXiv:2006.07237*.

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.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. arXiv:2002.06305.

Xin Dong, Shangyu Chen, and Sinno Pan. 2017. Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *International Conference on Learning Representations*.

Jeffrey L Elman. 1993. Learning and development in neural networks: The importance of starting small. *Cognition*, 48(1):71–99.

Angela Fan, Edouard Grave, and Armand Joulin. 2020. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Matthias Feurer, Katharina Eggensperger, Stefan Falkner, Marius Lindauer, and Frank Hutter. 2020. Auto-sklearn 2.0: The next generation. *arXiv preprint arXiv:2007.04074*, 24.

Matthias Feurer, Aaron Klein, Katharina Eggensperger, Jost Springenberg, Manuel Blum, and Frank Hutter. 2015. Efficient and robust automated machine learning. *Advances in neural information processing systems*, 28.

Yarin Gal, Riashat Islam, and Zoubin Ghahramani. 2017. Deep bayesian active learning with image data. In *International Conference on Machine Learning*, pages 1183–1192. PMLR.

Trevor Gale, Erich Elsen, and Sara Hooker. 2019. The state of sparsity in deep neural networks. *arXiv preprint arXiv:1902.09574*.

Tao Ge and Furu Wei. 2022. EdgeFormer: A Parameter-Efficient Transformer for On-Device Seq2seq Generation. *arXiv preprint arXiv:2202.07959*.

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *arXiv preprint arXiv:1907.06347*.

Mitchell Gordon, Kevin Duh, and Nicholas Andrews. 2020. Compressing BERT: Studying the effects of weight pruning on transfer learning. In *Proceedings of the 5th Workshop on Representation Learning for NLP*, pages 143–155, Online. Association for Computational Linguistics.

Albert Gu, Karan Goel, and Christopher Re. 2022. Efficiently modeling long sequences with structured state spaces. In *International Conference on Learning Representations*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided non-parametric neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4884–4896, Online. Association for Computational Linguistics.

Tae Jun Ham, Sung Jun Jung, Seonghak Kim, Young H. Oh, Yeonhong Park, Yoonho Song, Jung-Hun Park, Sanghee Lee, Kyoung Park, Jae W. Lee, and Deog-Kyoon Jeong. 2020. A^3: Accelerating Attention Mechanisms in Neural Networks with Approximation. In *2020 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pages 328–341. ISSN: 2378-203X.

Tae Jun Ham, Yejin Lee, Seong Hoon Seo, Soosung Kim, Hyunji Choi, Sung Jun Jung, and Jae W. Lee. 2021. ELSA: Hardware-Software Co-design for Efficient, Lightweight Self-Attention Mechanism in Neural Networks. In *2021 ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA)*, pages 692–705. ISSN: 2575-713X.

Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. 2022. Fastermoe: Modeling and optimizing training of large-scale dynamic pre-trained models. In *Proceedings of the 27th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP '22, page 120–134, New York, NY, USA. Association for Computing Machinery.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021a. Efficient Nearest Neighbor Language Models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714.

Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking ImageNet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2021b. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*.

Peter Henderson, Jieru Hu, Joshua Romoff, Emma Brunskill, Dan Jurafsky, and Joelle Pineau. 2020. Towards the systematic reporting of the energy and carbon footprints of machine learning. *Journal of Machine Learning Research*, 21(248):1–43.

Daniel Hershcovich, Nicolas Webersinke, Mathias Kraus, Julia Anna Bingler, and Markus Leippold. 2022. Towards climate awareness in nlp research. *arXiv preprint arXiv:2205.05071*.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network.

Torsten Hoefler, Dan Alistarh, Tal Ben-Nun, Nikoli Dryden, and Alexandra Peste. 2021. Sparsity in deep learning: Pruning and growth for efficient inference and training in neural networks. *Journal of Machine Learning Research*, 22(241):1–124.

Sara Hooker. 2021. The hardware lottery. *Communications of the ACM*, 64:58–65.

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *International Conference on Machine Learning*.

Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv:1704.04861 [cs]*. ArXiv: 1704.04861.

C-H Hsu, W-C Feng, and Jeremy S Archuleta. 2005. Towards efficient supercomputing: A quest for the right metric. In *19th IEEE International Parallel and Distributed Processing Symposium*, pages 8–pp. IEEE.

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*.

Zhiqi Huang, Lu Hou, Lifeng Shang, Xin Jiang, Xiao Chen, and Qun Liu. 2021. GhostBERT: Generate More Features with Cheap Operations for BERT. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6512–6523, Online. Association for Computational Linguistics.

Itay Hubara, Yury Nahshan, Yair Hanani, Ron Banner, and Daniel Soudry. 2021. Accurate post training quantization with small calibration sets. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 4466–4475. PMLR.

Forrest Iandola, Albert Shaw, Ravi Krishna, and Kurt Keutzer. 2020. SqueezeBERT: What can computer vision teach NLP about efficient neural networks? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 124–135, Online. Association for Computational Linguistics.

Eugenia Iofinova, Alexandra Peste, Mark Kurtz, and Dan Alistarh. 2022. How well do sparse imagenet models transfer? In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12266–12276.

Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International conference on machine learning*, pages 4651–4664. PMLR.

Kevin Jamieson and Ameet Talwalkar. 2016. Non-stochastic best arm identification and hyperparameter optimization. In *Artificial intelligence and statistics*, pages 240–248. PMLR.

William Stanley Jevons. 1866. *The Coal Question; An Inquiry Concerning the Progress of the Nation, and the Probable Exhaustion of Our Coal Mines*. Macmillan & Co. London.

Tianchu Ji, Shraddhan Jain, Michael Ferdman, Peter Milder, H. Andrew Schwartz, and Niranjan Balasubramanian. 2021. On the Distribution, Sparsity, and Inference-time Quantization of Attention Values in Transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4147–4157, Online. Association for Computational Linguistics.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Dhiraj Kalamkar, Dheevatsa Mudigere, Naveen Mellempudi, Dipankar Das, Kunal Banerjee, Sasikanth Avancha, Dharma Teja Vooturi, Nataraj Jammalamadaka, Jianyu Huang, Hector Yuen, Jiyan Yang, Jongsoo Park, Alexander Heinecke, Evangelos Georganas, Sudarshan Srinivasan, Abhisek Kundu, Misha Smelyanskiy, Bharat Kaul, and Pradeep Dubey. 2019. A Study of BFLOAT16 for Deep Learning Training. *arXiv preprint arXiv:1905.12322v3*.

Siddharth Karamcheti, Ranjay Krishna, Li Fei-Fei, and Christopher Manning. 2021. Mind your outliers! investigating the negative impact of outliers on active learning for visual question answering. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7265–7281, Online. Association for Computational Linguistics.

Rabeeh Karimi Mahabadi, James Henderson, and Sebastian Ruder. 2021. Compacter: Efficient low-rank hypercomplex adapter layers. In *Advances in Neural Information Processing Systems*, volume 34.

Angelos Katharopoulos, Apoorv Vyas, Nikolaos Pappas, and François Fleuret. 2020. Transformers are rnns: Fast autoregressive transformers with linear attention. In *International Conference on Machine Learning*, pages 5156–5165. PMLR.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proccedings of International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *International Conference on Learning Representations*.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2021. I-bert: Integer-only bert quantization. *International Conference on Machine Learning*.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Andreas Kirsch, Joost Van Amersfoort, and Yarin Gal. 2019. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. *Advances in neural information processing systems*, 32.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. 2020. Reformer: The efficient transformer. In *International Conference on Learning Representations*.

Jan-Christoph Klie, Bonnie Webber, and Iryna Gurevych. 2022. Annotation error detection: Analyzing the past and present for a more coherent future. *arXiv preprint arXiv:2206.02280*.

Olga Kovaleva, Alexey Romanov, Anna Rogers, and Anna Rumshisky. 2019. Revealing the dark secrets of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4365–4374, Hong Kong, China. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

M Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. *Advances in neural information processing systems*, 23.

Imad Lakim, Ebtesam Almazrouei, I. Abualhaol, Mérouane Debbah, and Julien Launay. 2022. A holistic assessment of the carbon footprint of noor, a very large arabic language model. In *Proc. of BigScience*.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2019. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Yann LeCun, John Denker, and Sara Solla. 1989. Optimal brain damage. In *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann.

Ji-Ung Lee, Jan-Christoph Klie, and Iryna Gurevych. 2022a. Annotation Curricula to Implicitly Train Non-Expert Annotators. *Computational Linguistics*, 48(2):343–373.

Ji-Ung Lee, Christian M. Meyer, and Iryna Gurevych. 2020. Empowering Active Learning to Jointly Optimize System and User Demands. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4233–4247, Online. Association for Computational Linguistics.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022b. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8424–8445, Dublin, Ireland. Association for Computational Linguistics.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. 2022. FNet: Mixing tokens with Fourier transforms. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.

Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim

Krikun, Noam Shazeer, and Zhifeng Chen. 2021. {GS}hard: Scaling giant models with conditional computation and automatic sharding. In *International Conference on Learning Representations*.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3045–3059, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020a. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

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

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110*.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-Tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. 2020. A system for massively parallel hyperparameter tuning. *Proceedings of Machine Learning and Systems*, 2:230–246.

Qin Li, Xiaofan Zhang, Jinjun Xiong, Wen-Mei Hwu, and Deming Chen. 2021. Efficient Methods for Mapping Neural Machine Translator on FPGAs. *IEEE Transactions on Parallel and Distributed Systems*, 32(7):1866–1877. Conference Name: IEEE Transactions on Parallel and Distributed Systems.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4582–4597, Online. Association for Computational Linguistics.

Opher Lieber, Or Sharir, Barak Lenz, and Yoav Shoham. 2021. Jurassic-1: Technical details and evaluation. Technical report, AI21 Labs.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. 2022. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23:54–1.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *arXiv preprint arXiv:2205.05638*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2021a. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *arXiv preprint arXiv:2107.13586*.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Xiangyang Liu, Tianxiang Sun, Junliang He, Jiawen Wu, Lingling Wu, Xinyu Zhang, Hao Jiang, Zhao Cao, Xuanjing Huang, and Xipeng Qiu. 2022b. Towards efficient NLP: A standard evaluation and a strong baseline. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3288–3303, Seattle, United States. Association for Computational Linguistics.

Xueqing Liu and Chi Wang. 2021. An empirical study on hyperparameter optimization for fine-tuning pre-trained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2286–2300, Online. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. arXiv:1907.11692.

Zejian Liu, Gang Li, and Jian Cheng. 2021b. Hardware Acceleration of Fully Quantized BERT for Efficient Natural Language Processing. *arXiv:2103.02800 [cs]*. ArXiv: 2103.02800.

David Lowell, Zachary C. Lipton, and Byron C. Wallace. 2019. Practical obstacles to deploying active learning. In *Proceedings of the 2019 Conference on*

Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 21–30, Hong Kong, China. Association for Computational Linguistics.

Siyuan Lu, Meiqi Wang, Shuang Liang, Jun Lin, and Zhongfeng Wang. 2020. Hardware accelerator for multi-head attention and position-wise feed-forward in the transformer. In *2020 IEEE 33rd International System-on-Chip Conference (SOCC)*, pages 84–89. IEEE.

Sasha Luccioni, Victor Schmidt, Alexandre Lacoste, and Thomas Dandres. 2019. Quantifying the carbon emissions of machine learning. In *NeurIPS 2019 Workshop on Tackling Climate Change with Machine Learning*.

Pedro Martins, Zita Marinho, and Andre Martins. 2022a. Efficient machine translation domain adaptation. In *Proceedings of the 1st Workshop on Semi-parametric Methods in NLP: Decoupling Logic from Knowledge*, pages 23–29, Dublin, Ireland and Online. Association for Computational Linguistics.

Pedro Martins, Zita Marinho, and André FT Martins. 2022b. Chunk-based nearest neighbor machine translation. *arXiv preprint arXiv:2205.12230*.

Pedro Henrique Martins, Zita Marinho, and Andre Martins. 2022c. ∞-former: Infinite memory transformer. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5468–5485, Dublin, Ireland. Association for Computational Linguistics.

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2022. Fast nearest neighbor machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 555–565, Dublin, Ireland. Association for Computational Linguistics.

Paul Michel, Omer Levy, and Graham Neubig. 2019. Are Sixteen Heads Really Better than One? In *Advances in Neural Information Processing Systems*, volume 32, pages 14014–14024. Curran Associates, Inc.

Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Erich Elsen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, and Hao Wu. 2018. Mixed Precision Training. In *International Conference on Learning Representations*.

Swaroop Mishra and Bhavdeep Singh Sachdeva. 2020. Do we need to create big datasets to learn a task? In *Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing*, pages 169–173, Online. Association for Computational Linguistics.

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget? *arXiv preprint arXiv:2205.10828*.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? *Advances in neural information processing systems*, 33:512–523.

Curtis G Northcutt, Anish Athalye, and Jonas Mueller. 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Amandalynne Paullada, Inioluwa Deborah Raji, Emily M. Bender, Emily Denton, and Alex Hanna. 2021. Data and its (dis)contents: A survey of dataset development and use in machine learning research. *Patterns*, 2(11):100336.

Hao Peng, Nikolaos Pappas, Dani Yogatama, Roy Schwartz, Noah Smith, and Lingpeng Kong. 2020. Random feature attention. In *International Conference on Learning Representations*.

Ben Peters and André F. T. Martins. 2021. Smoothing and shrinking the sparse Seq2Seq search space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2642–2654, Online. Association for Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Fabio Petroni, Tim Rocktäschel, Sebastian Riedel, Patrick Lewis, Anton Bakhtin, Yuxiang Wu, and Alexander Miller. 2019. Language models as knowledge bases? In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2463–2473, Hong Kong, China. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. 2019. Competence-based curriculum learning for neural machine translation. 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 1162–1172, Minneapolis, Minnesota. Association for Computational Linguistics.

Gabriele Prato, Ella Charlaix, and Mehdi Rezagholizadeh. 2020. Fully quantized transformer for machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1–14, Online. Association for Computational Linguistics.

Ofir Press, Noah A. Smith, and Mike Lewis. 2021. Shortformer: Better language modeling using shorter inputs. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5493–5505, Online. Association for Computational Linguistics.

Lucas Høyberg Puvis de Chavannes, Mads Guldborg Kjeldgaard Kongsbak, Timmie Rantzau, and Leon Derczynski. 2021. Hyperparameter power impact in transformer language model training. In *Proceedings of the Second Workshop on Simple and Efficient Natural Language Processing*, pages 96–118.

Zheng Qu, Liu Liu, Fengbin Tu, Zhaodong Chen, Yufei Ding, and Yuan Xie. 2022. DOTA: detect and omit weak attentions for scalable transformer acceleration. In *Proceedings of the 27th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, ASPLOS 2022, pages 14–26, New York, NY, USA. Association for Computing Machinery.

Jerry Quinn and Miguel Ballesteros. 2018. Pieces of eight: 8-bit neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 3 (Industry Papers)*, pages 114–120, New Orleans - Louisiana. Association for Computational Linguistics.

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Jack W Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, et al. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446*.

Jack W. Rae, Anna Potapenko, Siddhant M. Jayakumar, Chloe Hillier, and Timothy P. Lillicrap. 2020. Compressive transformers for long-range sequence modelling. In *International Conference on Learning Representations*.

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.

Samyam Rajbhandari, Conglong Li, Zhewei Yao, Minjia Zhang, Reza Yazdani Aminabadi, Ammar Ahmad Awan, Jeff Rasley, and Yuxiong He. 2022. Deepspeed-moe: Advancing mixture-of-experts inference and training to power next-generation ai scale.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

Machel Reid, Edison Marrese-Taylor, and Yutaka Matsuo. 2021. Subformer: Exploring weight sharing for parameter efficiency in generative transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4081–4090, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Jie Ren, Samyam Rajbhandari, Reza Yazdani Aminabadi, Olatunji Ruwase, Shuangyan Yang, Minjia Zhang, Dong Li, and Yuxiong He. 2021a. {ZeRO-Offload}: Democratizing {Billion-Scale} Model Training. In *2021 USENIX Annual Technical Conference (USENIX ATC 21)*, pages 551–564.

Pengzhen Ren, Yun Xiao, Xiaojun Chang, Po-Yao Huang, Zhihui Li, Brij B. Gupta, Xiaojiang Chen, and Xin Wang. 2021b. A survey of deep active learning. *ACM Comput. Surv.*, 54(9).

Adithya Renduchintala, Denise Diaz, Kenneth Heafield, Xian Li, and Mona Diab. 2021. Gender bias amplification during speed-quality optimization in neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 99–109, Online. Association for Computational Linguistics.

Aurko Roy, Mohammad Saffar, Ashish Vaswani, and David Grangier. 2021. Efficient content-based sparse attention with routing transformers. *Transactions of the Association for Computational Linguistics*, 9:53–68.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2021. AdapterDrop: On the efficiency of adapters in transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7930–7946, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2020. Poor Man's BERT: Smaller and Faster Transformer Models. *arXiv:2004.03844 [cs]*. ArXiv: 2004.03844.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*. ArXiv: 1910.01108.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *International Conference on Learning Representations*.

Victor Sanh, Thomas Wolf, and Alexander Rush. 2020. Movement pruning: Adaptive sparsity by fine-tuning. In *Advances in Neural Information Processing Systems*, volume 33, pages 20378–20389. Curran Associates, Inc.

Chinnadhurai Sankar, Sujith Ravi, and Zornitsa Kozareva. 2021. ProFormer: Towards On-Device LSH Projection Based Transformers. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2823–2828, Online. Association for Computational Linguistics.

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.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020a. Green AI. *Communications of the ACM (CACM)*, 63(12):54–63.

Roy Schwartz, Gabriel Stanovsky, Swabha Swayamdipta, Jesse Dodge, and Noah A. Smith. 2020b. The right tool for the job: Matching model and instance complexities. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6640–6651, Online. Association for Computational Linguistics.

Ozan Sener and Silvio Savarese. 2018. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*.

Burr Settles. 2012. *Active Learning*, volume 18 of *Synthesis Lectures on Artificial Intelligence and Machine Learning*. Morgan & Claypool.

Burr Settles, Mark Craven, and Lewis Friedland. 2008. Active learning with real annotation costs. In *Proceedings of the NIPS workshop on cost-sensitive learning (Vol. 1)*.

Noam Shazeer, *Azalia Mirhoseini, *Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. In *International Conference on Learning Representations*.

Sheng Shen, Zhen Dong, Jiayu Ye, Linjian Ma, Zhewei Yao, Amir Gholami, Michael W. Mahoney, and Kurt Keutzer. 2020. Q-BERT: Hessian Based Ultra Low Precision Quantization of BERT. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):8815–8821. Number: 05.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Andrew Silva, Pradyumna Tambwekar, and Matthew Gombolay. 2021. Towards a comprehensive understanding and accurate evaluation of societal biases in pre-trained transformers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2383–2389, Online. Association for Computational Linguistics.

Jasper Snoek, Hugo Larochelle, and Ryan P Adams. 2012. Practical Bayesian optimization of machine learning algorithms. *Advances in neural information processing systems*, 25.

Pierre Stock, Angela Fan, Benjamin Graham, Edouard Grave, Rémi Gribonval, Herve Jegou, and Armand Joulin. 2021. Training with quantization noise for extreme model compression. In *International Conference on Learning Representations*.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a compact task-agnostic BERT for resource-limited devices. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2158–2170, Online. Association for Computational Linguistics.

Yi-Lin Sung, Varun Nair, and Colin Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*.

Swabha Swayamdipta, Roy Schwartz, Nicholas Lourie, Yizhong Wang, Hannaneh Hajishirzi, Noah A.

Smith, and Yejin Choi. 2020. Dataset cartography: Mapping and diagnosing datasets with training dynamics. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9275–9293, Online. Association for Computational Linguistics.

Thierry Tambe, Coleman Hooper, Lillian Pentecost, En-Yu Yang, Marco Donato, Victor Sanh, Alexander M. Rush, David Brooks, and Gu-Yeon Wei. 2021. EdgeBERT: Optimizing On-Chip Inference for Multi-Task NLP. *arXiv:2011.14203 [cs]*. ArXiv: 2011.14203 version: 3.

Min Tang, Xiaoqiang Luo, and Salim Roukos. 2002. Active learning for statistical natural language parsing. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 120–127, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2020. Efficient transformers: A survey. *ACM Computing Surveys (CSUR)*.

Yi Tay, Shuohang Wang, Anh Tuan Luu, Jie Fu, Minh C. Phan, Xingdi Yuan, Jinfeng Rao, Siu Cheung Hui, and Aston Zhang. 2019. Simple and effective curriculum pointer-generator networks for reading comprehension over long narratives. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4922–4931, Florence, Italy. Association for Computational Linguistics.

Kale-ab Tessera, Sara Hooker, and Benjamin Rosman. 2021. Keep the gradients flowing: Using gradient flow to study sparse network optimization. *arXiv preprint arXiv:2102.01670*.

Neil C Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F Manso. 2020. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558*.

Marcos Treviso, António Góis, Patrick Fernandes, Erick Fonseca, and Andre Martins. 2022. Predicting attention sparsity in transformers. In *Proceedings of the Sixth Workshop on Structured Prediction for NLP*, pages 67–81, Dublin, Ireland. Association for Computational Linguistics.

Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov. 2019. Analyzing Multi-Head Self-Attention: Specialized Heads Do the Heavy Lifting, the Rest Can Be Pruned. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5797–5808, Florence, Italy. Association for Computational Linguistics.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1074–1080, Online. Association for Computational Linguistics.

Hanrui Wang, Zhekai Zhang, and Song Han. 2021a. SpAtten: Efficient Sparse Attention Architecture with Cascade Token and Head Pruning. In *2021 IEEE International Symposium on High-Performance Computer Architecture (HPCA)*, pages 97–110. ISSN: 2378-203X.

Shuhe Wang, Jiwei Li, Yuxian Meng, Rongbin Ouyang, Guoyin Wang, Xiaoya Li, Tianwei Zhang, and Shi Zong. 2021b. Faster nearest neighbor machine translation. *arXiv preprint arXiv:2112.08152*.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022b. Emergent Abilities of Large Language Models.

Ying Wei, Peilin Zhao, and Junzhou Huang. 2021. Meta-learning hyperparameter performance prediction with neural processes. In *International Conference on Machine Learning*, pages 11058–11067. PMLR.

Alexander Wettig, Tianyu Gao, Zexuan Zhong, and Danqi Chen. 2022. Should you mask 15% in masked language modeling? *arXiv preprint arXiv:2202.08005*.

Xiaoxia Wu, Zhewei Yao, Minjia Zhang, Conglong Li, and Yuxiong He. 2022. Extreme compression for pre-trained transformers made simple and efficient. *arXiv preprint arXiv:2206.01859*.

Zhanghao Wu*, Zhijian Liu*, Ji Lin, Yujun Lin, and Song Han. 2020. Lite transformer with long-short range attention. In *International Conference on Learning Representations*.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251, Online. Association for Computational Linguistics.

Benfeng Xu, Licheng Zhang, Zhendong Mao, Quan Wang, Hongtao Xie, and Yongdong Zhang. 2020. Curriculum learning for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6095–6104, Online. Association for Computational Linguistics.

Canwen Xu, Wangchunshu Zhou, Tao Ge, Ke Xu, Julian McAuley, and Furu Wei. 2021. Beyond preserved accuracy: Evaluating loyalty and robustness of BERT compression. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10653–10659, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Guangxuan Xu and Qingyuan Hu. 2022. Can model compression improve nlp fairness. *arXiv preprint arXiv:2201.08542*.

Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. 2021. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34.

Greg Yang and Etai Littwin. 2021. Tensor programs iib: Architectural universality of neural tangent kernel training dynamics. In *International Conference on Machine Learning*, pages 11762–11772. PMLR.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive Semiparametric Language Models. *Transactions of the Association for Computational Linguistics*.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7935–7948, Online. Association for Computational Linguistics.

Michelle Yuan, Patrick Xia, Chandler May, Benjamin Van Durme, and Jordan Boyd-Graber. 2022. Adapting coreference resolution models through active learning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7533–7549, Dublin, Ireland. Association for Computational Linguistics.

A. H. Zadeh, I. Edo, O. M. Awad, and A. Moshovos. 2020. GOBO: Quantizing Attention-Based NLP Models for Low Latency and Energy Efficient Inference. In *2020 53rd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO)*, pages 811–824.

Ali Hadi Zadeh, Mostafa Mahmoud, Ameer Abdelhadi, and Andreas Moshovos. 2022. Mokey: enabling narrow fixed-point inference for out-of-the-box floating-point transformer models. In *Proceedings of the 49th Annual International Symposium on Computer Architecture*, ISCA '22, pages 888–901, New York, NY, USA. Association for Computing Machinery.

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8Bit BERT. *arXiv:1910.06188 [cs]*. ArXiv: 1910.06188.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. 2020. Big bird: Transformers for longer sequences. *Advances in Neural Information Processing Systems*, 33:17283–17297.

Runzhe Zhan, Xuebo Liu, Derek F Wong, and Lidia S Chao. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14310–14318.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. TernaryBERT: Distillation-aware Ultra-low Bit BERT. *arXiv:2009.12812 [cs, eess]*. ArXiv: 2009.12812.

Xuan Zhang and Kevin Duh. 2020. Reproducible and efficient benchmarks for hyperparameter optimization of neural machine translation systems. *Transactions of the Association for Computational Linguistics*, 8:393–408.

Xuan Zhang, Pamela Shapiro, Gaurav Kumar, Paul McNamee, Marine Carpuat, and Kevin Duh. 2019. Curriculum learning for domain adaptation in neural machine translation. 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 1903–1915, Minneapolis, Minnesota. Association for Computational Linguistics.

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pretrained neural machine translation models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 9652–9659.

Yikai Zhou, Baosong Yang, Derek F. Wong, Yu Wan, and Lidia S. Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944, Online. Association for Computational Linguistics.

Qingqing Zhu, Xiuying Chen, Pengfei Wu, JunFei Liu, and Dongyan Zhao. 2021. Combining curriculum learning and knowledge distillation for dialogue generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 1284–1295, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Lucas Zimmer, Marius Lindauer, and Frank Hutter. 2021. Auto-pytorch: Multi-fidelity metalearning

for efficient and robust autodl. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(9):3079–3090.