Efficient Methods for Natural Language Processing: A Survey

Marcos Treviso[1*], **Ji-Ung Lee**[2*], **Tianchu Ji**[3*], **Betty van Aken**[4], **Qingqing Cao**[5], **Manuel R. Ciosici**[6], **Michael Hassid**[7], **Kenneth Heafield**[8], **Sara Hooker**[9], **Colin Raffel**[10], **Pedro H. Martins**[1,11], **André F. T. Martins**[1,11], **Jessica Zosa Forde**[12], **Peter Milder**[3], **Edwin Simpson**[13], **Noam Slonim**[14], **Jesse Dodge**[15], **Emma Strubell**[15,16], **Niranjan Balasubramanian**[3], **Leon Derczynski**[5,17], **Iryna Gurevych**[2], **Roy Schwartz**[7]

[1]IST/U. of Lisbon and Instituto de Telecomunicações, Portugal,
[2]Technical University of Darmstadt, Germany, [3]Stony Brook University, USA,
[4]Berliner Hochschule für Technik, Germany, [5]University of Washington, USA,
[6]University of Southern California, USA, [7]The Hebrew University of Jerusalem, Israel,
[8]University of Edinburgh, UK, [9]Cohere For AI, USA,
[10]University of North Carolina at Chapel Hill, USA, [11]Unbabel, Portugal, [12]Brown University, USA,
[13]University of Bristol, UK, [14]IBM Research, Israel, [15]Allen Institute for AI, USA,
[16]Carnegie Mellon University, USA, [17]IT University of Copenhagen, Denmark

Abstract

Recent work in natural language processing (NLP) has yielded appealing results from scaling model parameters and training data; however, using only scale to improve performance means that resource consumption also grows. Such resources include data, time, storage, or energy, all of which are naturally limited and unevenly distributed. This motivates research into *efficient* methods that require fewer resources to achieve similar results. This survey synthesizes and relates current methods and findings in efficient NLP. We aim to provide both guidance for conducting NLP under limited resources, and point towards promising research directions for developing more efficient methods.

1 Introduction

Scaling has become a key ingredient in achieving state-of-the-art performance in NLP (Figure 1), as recent research suggests that some capabilities only emerge once models grow beyond a certain size (Wei et al., 2022b). However, despite the merits of scaling, it poses key challenges to making these breakthroughs accessible in resource-constrained environments (Ahmed and Wahed, 2020), in having a non-negligible environmental impact (Strubell et al., 2019; Schwartz et al., 2020a; Derczynski, 2020; Patterson et al., 2021; Wu et al., 2022a), and in complying with hardware constraints (Thompson et al., 2020). To tackle these limitations, there has been renewed focus around research that seeks to improve model *efficiency*.

Definition Efficiency is characterized by the relationship between resources going into a system and its output, with a more efficient system producing the same output with fewer resources. Schwartz et al. (2020a) formalize efficiency as the cost of a model in relation to the results it produces: $\mathrm{Cost}(R) \propto E \cdot D \cdot H$, i.e., the $\mathrm{Cost}(\cdot)$ of producing a certain NLP (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 (D)ataset, and (3) the number of training runs required for (H)yperparameter tuning. Here we take a different approach, and consider the role that efficiency plays across the different steps in the NLP pipeline, by providing a detailed overview of efficiency methods specific to NLP (Figure 2).

Scope of this Survey We address this work to two groups of readers: (1) Researchers from all fields of NLP working with limited resources; and (2) Researchers interested in improving the state of the art of efficient methods in NLP. Each section concludes with a discussion of limitations, open challenges, and possible future directions of the presented methods. We start by discussing methods to increase *data* efficiency (Section 2), and continue with methods related to *model design* (Section 3). We then consider efficient methods

[*]Equal contribution. marcos.treviso@tecnico.pt.

Transactions of the Association for Computational Linguistics, vol. 11, pp. 826–860, 2023. https://doi.org/10.1162/tacl_a_00577
Action Editor: Xavier Carreras. Submission batch: 10/2022; Revision batch: 1/2023; Published 7/2023.

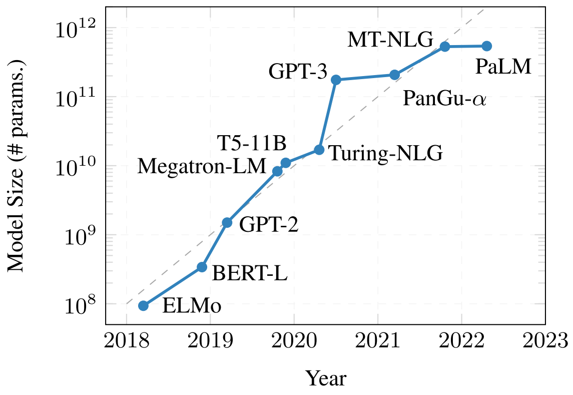

Figure 1: Exponential growth in the number of parameters in pre-trained language models. Adapted from Lakim et al. (2022).

for the two typical training setups in modern NLP: *pre-training* (Section 4) and *fine-tuning* (Section 5). We then discuss methods for making *inference* more efficient (Section 6). While we mainly focus on algorithmic approaches, we provide appropriate pointers regarding *hardware* that are connected to the scale at which we expect to deploy a model (Section 7). We then discuss how to quantify efficiency and what factors to consider during *evaluation* (Section 8), and, finally, how to efficiently decide upon the *best suited model* (Section 9).

To guide the reader, Figure 3 presents a typology of efficient NLP methods considered in this survey.

2 Data

Data efficiency is improved by using fewer training instances, or by making better use of available instances. Fixed compute budgets motivate balancing model size and training data size, especially during pre-training (Hoffmann et al., 2022).

2.1 Filtering

Improving *data quality* can boost performance while reducing training costs during pre-training and fine-tuning. For instance, Lee et al. (2022b) showed that removing duplicates in pre-training increases training efficiency, giving equal or even better model performance compared to using all data. Zhang et al. (2022) used MinhashLSH (Leskovec et al., 2020) to remove duplicates while developing OPT. De-duplication can lead to substantially reduced computation cost, especially in cases with abundant pre-training data but limited compute budget (Hoffmann et al., 2022).

Similar observations have been made for fine-tuning. For instance, Mishra and Sachdeva (2020) found—via adversarial filtering (Zellers et al., 2018)—a subset of only ~2% of the SNLI data (Bowman et al., 2015) that leads to performance comparable to using the full corpus. While such filtering approaches are useful for mitigating biases (Le Bras et al., 2020), they may not always serve as tools to filter existing datasets, as these often suffer from insufficient training data.

2.2 Active Learning

Active learning aims to reduce the number of training instances. In contrast to filtering, it is applied during data collection (instead of after) to only annotate the most helpful or useful instances for training (Settles, 2012; Ren et al., 2021b). To assess usefulness of an instance without knowing its actual label, one can use the model *uncertainty*—assuming that labeling instances with the highest uncertainty is most helpful (Lewis and Gale, 1994; Tang et al., 2002; Gal et al., 2017; Yuan et al., 2020); instance *representativeness*—to maximize diversity of sampled instances while avoiding outliers (Bodó et al., 2011; Sener and Savarese, 2018; Gissin and Shalev-Shwartz, 2019); or a combination of both criteria (Kirsch et al., 2019; Ash et al., 2020; Margatina et al., 2021; Siddiqui et al., 2021; Agarwal et al., 2022). Active learning has been successfully applied in machine translation (MT; Liu et al. 2018), language learning (Lee et al., 2020), entity linking (Klie et al., 2020), and coreference resolution (Li et al., 2020a; Yuan et al., 2022). Despite its advantages, some open questions make active learning difficult to apply in practice. It remains unclear how model-based sampling impacts the performance of models using architectures different from that in sampling (Lowell et al., 2019; Ein-Dor et al., 2020). Also, selecting ‘‘difficult’’ instances may increase annotation cost and difficulty (Settles et al., 2008; Lee et al., 2022a). Finally, it is prone to selection biases and can favor outliers (Cortes et al., 2008; Karamcheti et al., 2021).

2.3 Curriculum Learning

Curriculum learning aims to find a data ordering that reduces the number of training steps required to achieve a target performance (Elman,

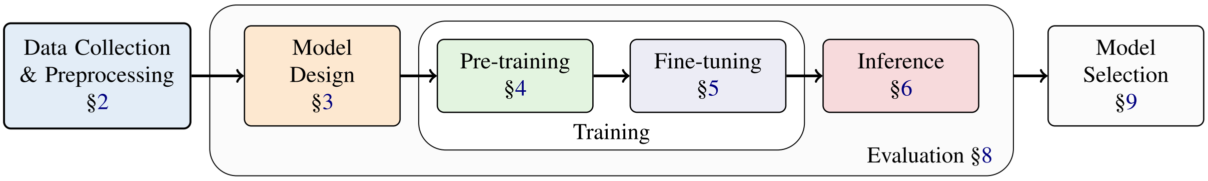

Figure 2: Schematic overview of the efficient NLP stages covered in this paper, starting with data collection and model design, followed by training and inference, and ending with evaluation and model selection. Notably, the training stage is divided into two parts: pre-training, which aims to learn generalizable parameters, and fine-tuning, which optimizes these parameters for specific downstream tasks.

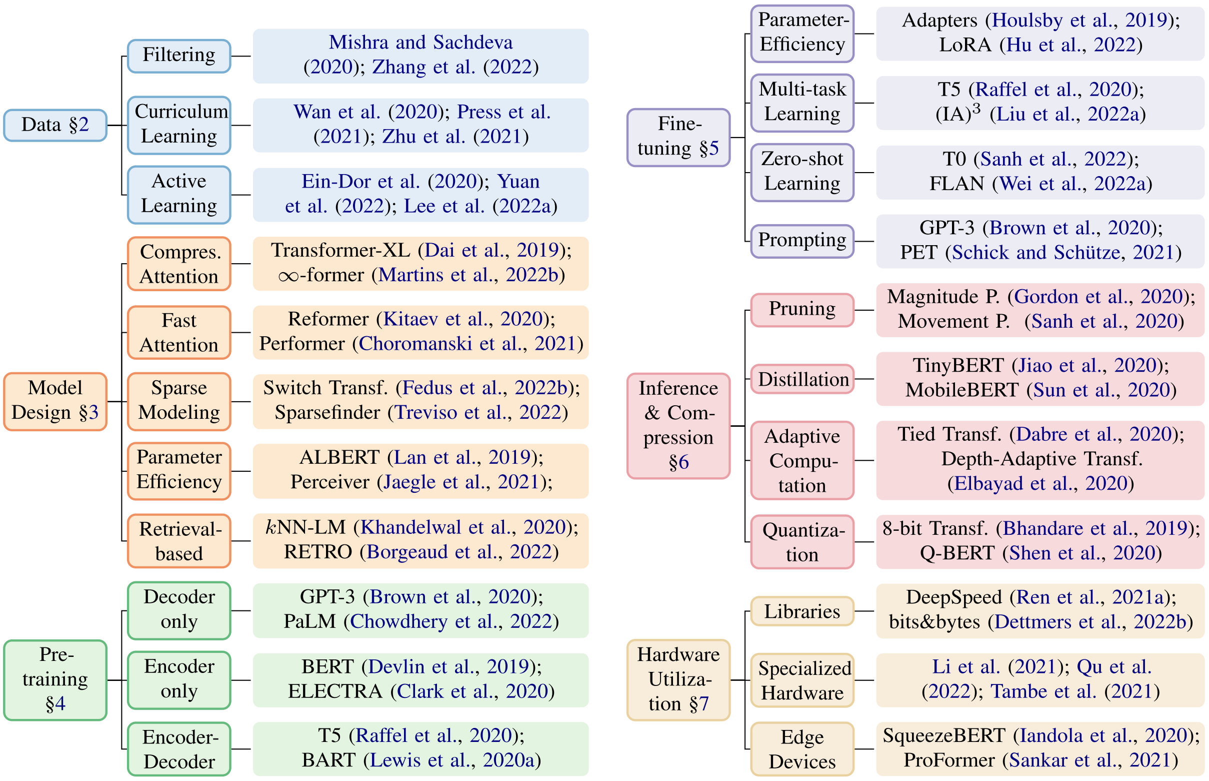
Data §2
Filtering
Mishra and Sachdeva (2020); Zhang et al. (2022)
Curriculum Learning
Wan et al. (2020); Press et al. (2021); Zhu et al. (2021)
Active Learning
Ein-Dor et al. (2020); Yuan et al. (2022); Lee et al. (2022a)
Model Design §3
Compres. Attention
Transformer-XL (Dai et al., 2019); ∞-former (Martins et al., 2022b)
Fast Attention
Reformer (Kitaev et al., 2020); Performer (Choromanski et al., 2021)
Sparse Modeling
Switch Transf. (Fedus et al., 2022b); Sparsefinder (Treviso et al., 2022)
Parameter Efficiency
ALBERT (Lan et al., 2019); Perceiver (Jaegle et al., 2021);
Retrieval-based
kNN-LM (Khandelwal et al., 2020); RETRO (Borgeaud et al., 2022)
Pre-training §4
Decoder only
GPT-3 (Brown et al., 2020); PaLM (Chowdhery et al., 2022)
Encoder only
BERT (Devlin et al., 2019); ELECTRA (Clark et al., 2020)
Encoder-Decoder
T5 (Raffel et al., 2020); BART (Lewis et al., 2020a)
Fine-tuning §5
Parameter-Efficiency
Adapters (Houlsby et al., 2019); LoRA (Hu et al., 2022)
Multi-task Learning
T5 (Raffel et al., 2020); (IA)^3 (Liu et al., 2022a)
Zero-shot Learning
T0 (Sanh et al., 2022); FLAN (Wei et al., 2022a)
Prompting
GPT-3 (Brown et al., 2020); PET (Schick and Schütze, 2021)
Inference & Compression §6
Pruning
Magnitude P. (Gordon et al., 2020); Movement P. (Sanh et al., 2020)
Distillation
TinyBERT (Jiao et al., 2020); MobileBERT (Sun et al., 2020)
Adaptive Computation
Tied Transf. (Dabre et al., 2020); Depth-Adaptive Transf. (Elbayad et al., 2020)
Quantization
8-bit Transf. (Bhandare et al., 2019); Q-BERT (Shen et al., 2020)
Hardware Utilization §7
Libraries
DeepSpeed (Ren et al., 2021a); bits&bytes (Dettmers et al., 2022b)
Specialized Hardware
Li et al. (2021); Qu et al. (2022); Tambe et al. (2021)
Edge Devices
SqueezeBERT (Iandola et al., 2020); ProFormer (Sankar et al., 2021)

Figure 3: Typology of efficient NLP methods.

1993; Bengio et al., 2009). This method does not reduce dataset size, but does improve its utilization. Hence, it is a common approach for improving training efficiency in both pre-training and fine-tuning. Many curriculum learning methods order instances by difficulty, using heuristics such as sentence length. This has yielded improvements for transformer pre-training (Press et al., 2021; Agrawal et al., 2021) as well as fine-tuning on tasks such as question answering (Tay et al., 2019), MT (Zhang et al., 2019), and others (Xu et al., 2020).

A major challenge in curriculum learning is determining *pace*, i.e., when to progress to more difficult instances. If not chosen carefully, curriculum learning can waste compute on ‘‘easy’’ instances. To tackle this, work has investigated adaptive ordering strategies based on current model state, called *self-paced learning* (Kumar et al., 2010). This has been successfully applied to improve performance in MT using model and data uncertainty (Wan et al., 2020; Zhou et al., 2020; Zhao et al., 2020), and in dialog generation with knowledge distillation (Zhu et al., 2021). However, self-paced learning involves large training costs, and disentangling instance ordering from factors such as optimizer choice and batch size is non-trivial (Dodge et al., 2020).

2.4 Estimating Data Quality

In an era of ever larger datasets, auditing and estimating the quality of data is increasingly challenging. Datasets frequently present high levels of noise and misaligned instances (Kreutzer et al., 2022). Estimating data quality encompasses research efforts which propose better uncertainty estimates (Baldock et al., 2021; D'souza et al., 2021; Ethayarajh et al., 2022) as well as analytical tools such as dataset cartography (Swayamdipta et al., 2020). Qualitative tools include documentation for datasets and model attributes (Gebru et al., 2021).

3 Model Design

Efficient model design covers architectural changes and adding new modules to accelerate training.

3.1 Improving Attention in Transformers

The transformer's self-attention mechanism has a quadratic dependency on sequence length which is not fully utilized by existing models (Hassid et al., 2022). To reduce computational costs, efficient attention mechanisms for long sequences have been proposed (Tay et al., 2022). Existing strategies include better using already-processed segments via recurrence to connect multiple segments (Dai et al., 2019), learning a network to compress a longer-term memory (Rae et al., 2020), separately modeling global and local attention (Ainslie et al., 2020), and modeling long inputs as a continuous-time signal (Martins et al., 2022b). Another line of research uses fixed attention patterns, where tokens attend to their immediate context (local attention) and possibly to a few global positions (global attention; Beltagy et al., 2020; Zaheer et al., 2020; Child et al., 2019). Compared to using the full self-attention matrix, such approaches can scale linearly with the input length.

Some methods learn attention sparsity patterns directly from data, e.g., by grouping tokens into buckets, leading to a more accurate yet more expensive approximation of the full attention matrix (Kitaev et al., 2020; Daras et al., 2020; Roy et al., 2021). Instead of seeking better attention patterns, some strategies modify the attention *mechanism* and derive low-rank approximations to the query-key matrices via reverse application of the kernel trick, resulting in linear time attention (Katharopoulos et al., 2020; Choromanski et al., 2021; Peng et al., 2020; Zhai et al., 2021). Recently, IO-aware attention mechanisms have been proposed, decreasing reads and writes to the attention matrix to GPU high-bandwidth memory (Dao et al., 2022b).

Despite various improvements in attention mechanisms, most of them struggle with very long sequences (Tay et al., 2021). S4 (Gu et al., 2022b), and its successors (Gupta et al., 2022; Mehta et al., 2023; Gu et al., 2022a), suggest an alternative to transformers that alleviates the short memory problem and the quadratic bottleneck cost of self-attention by discretizing state space representations through parameterization of the state matrix. More recently, Mega (Ma et al., 2023) replaced the multi-headed transformer attention mechanism with a single-headed mechanism that receives contextualized vectors from a multidimensional exponential moving average module, and then splits the input into multiple fixed-length chunks to reduce the computation cost. Both S4 and Mega strongly outperform attention-based methods on all tasks of the Long Range Arena benchmark (Tay et al., 2021), while increasing training speed by approximately 5x and reducing memory cost by about 15% when compared to a standard transformer. This success is attributed to their convolutional structure, which emphasizes nearby tokens and has a parameter count that grows sub-linearly with sequence length (Li et al., 2022b).

3.2 Sparse Modeling

To leverage sparsity for efficiency, many models follow the mixture-of-experts (MoE) concept (Jacobs et al., 1991; Shazeer et al., 2017; Fedus et al., 2022a), which routes computation through small subnetworks instead of passing the input through the entire model. Relevant works on this line include GShard (Lepikhin et al., 2021), Switch Transformer (Fedus et al., 2022b), and ST-MoE (Zoph et al., 2022), which replace the feed-forward layers in transformers with MoE layers. More recently, Rajbhandari et al. (2022) scaled transformers up by compressing and optimizing the usage of MoE. Overall, MoE models have been shown to achieve strong performance across several NLP tasks while reducing the overall resource consumption (Section 8). For instance, GLaM (Du et al., 2022) used only $\sim\frac{1}{3}$

of GPT-3's energy consumption (with additional hardware-based optimization), while Rajbhandari et al. (2022) reached a 5x reduction in terms of training cost. However, MoE models have also exhibited training instabilities in practice, and may require architecture-specific implementation (Zoph et al., 2022; Mustafa et al., 2022).

Another promising direction for exploiting sparse modeling is Sparsefinder (Treviso et al., 2022), which extends the Adaptively Sparse Transformer (Correia et al., 2019) to allow a more efficient attention mechanism by identifying beforehand the sparsity pattern returned by entmax attention—a sparse alternative to (dense) softmax attention (Peters et al., 2019). Finally, sparsity can also be induced via modularity, e.g., by encapsulating task-specific parameters (Ponti et al., 2022).

3.3 Parameter Efficiency

Methods that reduce parameter count can reduce computational costs and memory usage. One such approach is to share weights across layers of a model while maintaining the downstream task performance (Dehghani et al., 2019; Lan et al., 2019). Besides sharing weights, Perceiver (Jaegle et al., 2021) also minimizes the computational cost of self-attention on long sequences by mapping the input to a small latent vector. ALBERT (Lan et al., 2019) further uses matrix decomposition to reduce the size of the embedding layer, which is one of the largest consumers of model parameters. Finally, Reid et al. (2021) studied ways to share weights in transformers, showing that sharing only the middle layers of the model outperforms the alternatives.

3.4 Retrieval-Augmented Models

Parametric models can be combined with retrieval mechanisms for text generation, leading to semi-parametric models (Gu et al., 2018; Lewis et al., 2020b; Li et al., 2022a). This typically amounts to trading model size with the number of database entries. For instance, RETRO (Borgeaud et al., 2022) matched the performance of models 25 times larger by retrieving chunks of tokens from a 2 trillion token database. At inference time, the model retrieves tokens / phrases / sentences from a database, which are used by the model through a combination of probability distributions (Khandelwal et al., 2020), gating mechanisms (Yogatama et al., 2021), or attention (Borgeaud Borgeaud et al., 2022).

These models also have good generalization properties: By retrieving from domain-specific databases, they can be applied to new domains, reducing the need for domain-specific fine-tuning (Khandelwal et al., 2020, 2021). That is, having an explicit ''memory'' also allows retrieval-augmented models to be adapted post-training. Although they may yield slow running speeds since the retrieval time grows as the datastore scales, recent work proposed strategies to alleviate this, such as pruning the database (He et al., 2021), having smaller input-dependent databases (Meng et al., 2022), reducing the representation dimension (Martins et al., 2022a), and clustering data points (Wang et al., 2021b; Alon et al., 2022). In particular, Martins et al. (2022c) have shown that carefully constructing a database not only leads to better translations than fine-tuning, but can also reduce the total translation time (inference + online adaptation).

3.5 Model Design Considerations

Despite considerable advances, one major challenge is modeling long sequences in many real-world documents. For instance, sustainability reports have on average 243.5 pages (Manes-Rossi et al., 2018), which substantially exceeds the maximum length (16k tokens) found in Path-X from Long Range Arena (Tay et al., 2021). In fact, the ability of a model to handle longer sequences than those seen during training may depend on design choices, such as the attention mechanism (Dubois et al., 2020) and the positional encoding (Shaw et al., 2018; Press et al., 2022). The effect of this behavior when using transformers with sub-quadratic attention, sparse modeling approaches, or parameter efficient models is not yet well understood.

While sparse modeling approaches like MoE can substantially reduce inference and training costs, they require additional model parameters for retraining specialized modules and have instability issues during training (Zoph et al., 2022). Models that rely on built-in sparse transformations, such as entmax (Peters et al., 2019), have achieved strong results without stability issues, but have not yet fully realized competitive efficiency gains. Combining MoE with built-in sparse functions may be a promising research direction, e.g., by using entmax in the routing layer.

In retrieval-augmented models, the quality of the retrieval component is critical to performance, and the tradeoff between storing information in model parameters vs. external resources needs to be better understood, especially when deploying models in low-resource settings like edge devices. Finally, while new model designs improve efficiency through different means, further improvements can emerge from combining approaches, such as making MoE more efficient using quantization (Section 6.3) and using parameter-efficient models for distillation (Section 6.2).

4 Pre-training

Modern transfer learning approaches in NLP typically involve *pre-training* a model in a self-supervised fashion on large amounts of text before fine-tuning it on specific tasks (Section 5). Improving the pre-training procedure of a model can significantly reduce the cost of hyperparameter tuning and increase data efficiency for fine-tuning (Peters et al., 2018; He et al., 2019; Neyshabur et al., 2020).

4.1 Optimization Objective

The choice of the task can determine the success of the pre-trained model on downstream tasks. Left-to-right language models, such as GPT (Radford et al., 2019; Brown et al., 2020) and PaLM (Chowdhery et al., 2022), are trained with the *causal language modeling* (CLM) objective, which involves predicting the next token given a context. BERT (Devlin et al., 2019) uses a *masked language model* (MLM) task, which involves filling randomly masked tokens.

To make better use of available data, various masking strategies have been investigated. Masking objects and content words only rather than random tokens (Bitton et al., 2021), or masking more tokens (Wettig et al., 2022), has led to higher task performance and more efficient use of the available data. ELECTRA (Clark et al., 2020) and DeBERTa (He et al., 2023) tried *replaced token detection* (RTD), an objective that uses a small generator model to replace input tokens, and converges more quickly to better performance. A limitation of the MLM and RTD objectives is that they work with single token replacements. T5 (Raffel et al., 2020) and BART (Lewis et al., 2020a) overcome this by adopting a denoising *sequence-to-sequence* objective to pre-train an encoder-decoder model, allowing the decoder to predict a span of tokens for masked positions. In practice, this allows training on shorter sequences without losing task performance, which helps to reduce training costs.

4.2 Pre-training Considerations

Despite increases in the size of pre-trained models (cf. Figure 1), many pre-training efficiency gains come from improving model design (Section 3) and selection (Section 9) as well as making more efficient use of the available data (Section 2). These factors have had a greater impact on model performance than the pre-training objective itself (Alajrami and Aletras, 2022). However, pre-training is usually computationally expensive, requiring significant amounts of GPU memory and computational power (Rae et al., 2021), and may require large amounts of quality data, which can be difficult to acquire and curate (Kaplan et al., 2020). Surprisingly, as demonstrated by Chinchilla (Hoffmann et al., 2022), decreasing model size to account for the amount of available data not only leads to better performance, but also reduces computational cost and improves model applicability to downstream tasks. Continued focus on the role of data in efficient pre-training is a promising direction, such as recent work studying the role of (de-)duplication of examples in large-scale pre-training corpora (Lee et al., 2022b). While transformers have been the dominant architecture in pre-trained models, more efficient modeling methods such as state space representations and MoEs (Section 3.1) have the potential to overcome some challenges of pre-training transformers.

5 Fine-tuning

Fine-tuning refers to adapting a pre-trained model to a new downstream task. While some approaches explicitly aim to make the fine-tuning process more efficient, 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 model parameters on a downstream task. Hence, fine-tuning a pre-trained model on a new task creates an entirely new set of model parameters. If a model is fine-tuned on many tasks, the storage requirements can become onerous. Adapting a pre-trained model to downstream tasks by training a new classification layer and leaving the rest of the parameters fixed (a.k.a. feature extraction; Peters et al., 2018) updates dramatically fewer parameters than training the full model but has been shown to produce worse performance and has become less common (Devlin et al., 2019).

Several approaches have been proposed to adapt a model to a new task while only updating or adding a relatively small number of parameters—up to four orders of magnitude fewer parameters than full-model fine-tuning—without sacrificing (and in some cases improving) performance. Adapters (Houlsby et al., 2019; Bapna and Firat, 2019; Rebuffi et al., 2017; Pfeiffer et al., 2020) inject new trainable dense layers into a pre-trained model, while leaving the original model parameters fixed. They have recently been improved by the Compacter method (Karimi Mahabadi et al., 2021), which constructs the adapter parameter matrices through Kronecker products of low-rank matrices. While adapters can reduce training time due to a reduced number of trained parameters, and mitigate some deployment costs due to reduced storage requirements, one shortcoming is increased inference time due to more parameters (Rücklé et al., 2021). To mitigate this, Moosavi et al. (2022) proposed training an additional layer selector to only use adapter layers necessary for a given task.

As an alternative to adding new layers, parameter-efficiency can be achieved by directly modifying activations with learned vectors, either by concatenation (Liu et al., 2021a; Li and Liang, 2021; Lester et al., 2021), multiplication (Liu et al., 2022a), or addition (Ben Zaken et al., 2022). Two notable approaches are prefix-tuning (Li and Liang, 2021) and prompt-tuning (Lester et al., 2021), which fine-tune continuous prompts as an alternative to engineering discrete prompts (cf. Section 5.3). Although they are conceptually similar to adapters, He et al. (2022b) show that they are equivalent to a parallel insertion, whereas adapters are inserted sequentially. Alternatively, rather than adding new parameters or changing the computational graph, it is possible to make sparse (Sung et al., 2021; Guo et al., 2021) or low-rank (LoRA, Hu et al., 2022) updates. Finally, optimization can be performed in a low-dimensional subspace (Li et al., 2018), which leads to parameter-efficient updates (Aghajanyan et al., 2021b). Although low-rank approaches mitigate the issue of increased inference time, they require an additional optimization step to identify the best rank. To mitigate this, Valipour et al. (2022) proposed a dynamic solution that substantially reduces training time compared to LoRA. Lastly, Wang et al. (2022b) devised AdaMix to combine different parameter efficient fine-tuning techniques together via routing and showed that their approach can even outperform full fine-tuning.

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 fine-tuning on data from all downstream tasks of interest. Multi-task models can improve fine-tuning performance (Raffel et al., 2020; Aghajanyan et al., 2021a; Aribandi et al., 2022; Liu et al., 2022a). In certain cases, a multi-task model works on new tasks without any fine-tuning, also referred to as *zero-shot generalization* (Sanh et al., 2022; Wei et al., 2022a). Radford et al. (2017, 2019) and Brown et al. (2020) demonstrated that language models trained with an unsupervised objective can perform a variety of tasks out-of-the-box. While it can circumvent the need for fine-tuning, zero-shot ability depends on model size and only becomes competitive at a certain scale (Wei et al., 2022b).

5.3 Prompting

Inspired by models like GPT-3 (Brown et al., 2020), prompting refers to casting a task as a textual instruction to a language model (Liu et al., 2023). In general, prompts can be either crafted manually or automatically using fill-in templates for token, span, and sentence-level completion (Petroni et al., 2019; Brown et al., 2020; Shin et al., 2020). This makes prompting applicable to more challenging NLP tasks, such as QA, MT, and summarization (Schick and Schütze, 2021). Although prompting eliminates the need for any fine-tuning, identifying good prompts can be difficult (Liu et al., 2021a). Hence, recent work investigates the automated creation of suitable prompts, albeit with additional training cost (Bach et al., 2022).

5.4 Fine-Tuning Considerations

An emerging problem with large language models is the universally high cost of fully fine-tuning them (Chen et al., 2021). Although prompting (without fine-tuning) can alleviate this issue, designing prompts can be tedious—even with automated help. One promising direction for efficiently introducing new knowledge into models is to combine existing methods for efficient fine-tuning. This could involve methods such as that used by Karimi Mahabadi et al. (2022), who proposed task-specific adapters to avoid generating prompts, and achieved considerable speed ups while tuning under 1% of parameters. Another challenge in adopting large pre-trained models for fine-tuning is the complexity in interpreting the final model, due in part to the use transformers. To gain a better understanding of these models while still leveraging efficiency, a promising direction is to combine techniques such as sparse modeling and parameter-efficient methods (Correia et al., 2019; Treviso et al., 2022).

6 Inference and Compression

Inference involves computing a trained model's prediction for a given input. Inference can be made more efficient by accelerating the process for time efficiency (latency), or by compressing the model to reduce memory requirements.

6.1 Pruning

Proposed by LeCun et al. (1989), pruning removes irrelevant weights from a neural network to reduce computation, and furthermore, decreases memory capacity and bandwidth requirements. Pruning can be applied at different stages of the NLP pipeline (Figure 2). For instance, Gordon et al. (2020) found that up to $\sim$40% of BERT can be pruned at pre-training without affecting its performance. Others proposed pruning methods that work as regularizers and can be applied to pre-training and fine-tuning (Louizos et al., 2018; Wang et al., 2020b). Finally, work has investigated pruning during fine-tuning (Han et al., 2015; Sanh et al., 2020) or dynamically during inference (Fan et al., 2020).

Pruning was initially introduced at the individual weight level (unstructured pruning), but more recent approaches prune larger components of the network (structured pruning). Examples of the latter include removing attention heads (Voita et al., 2019; Michel et al., 2019), weak attention values (Ji et al., 2021; Qu et al., 2022), and even entire hidden layers (Dong et al., 2017; Sajjad et al., 2023). In particular, Xia et al. (2022) found that pruning all these components yields more accurate and efficient models. When comparing the two pruning approaches, unstructured pruning is often found to better preserve a model's performance (Gale et al., 2019; Ahia et al., 2021), but existing hardware often cannot exploit the resulting sparsity. In contrast, structured pruning methods often lead to a higher improvement in terms of inference speed (Hoefler et al., 2021). The increasing popularity of pruning methods has further raised the question of how to quantify and compare them (Gale et al., 2019; Blalock et al., 2020; Tessera et al., 2021; Hoefler et al., 2021) and motivated work that combines pruning with other efficiency methods such as adapters (Rücklé et al., 2021) and distillation (Zafrir et al., 2021).

While early pruning (e.g., during pre-training) can further reduce training costs, it increases the risk of over-pruning: removing nodes essential for downstream task performance (Gordon et al., 2020). Although this can be mitigated by ''regrowing'' pruned weights (Mostafa and Wang, 2019), this increases training costs. Other pruning downsides include additional costs for hyperparameter tuning such as the number of preserved weights.

6.2 Knowledge Distillation

The process of knowledge distillation uses supervision signals from a large (teacher) model to train a smaller (student) model (Hinton et al., 2015), and often leads to the student outperforming a similarly sized model trained without this supervision. While early work focused on distilling task-specific models (Kim and Rush, 2016), recent work focuses 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; Gou et al., 2021). The downsides of distillation include the added cost of tuning student hyperparameters and the potential for reduced performance and generalization capability (Stanton et al., 2021). Recently, Zhu et al. (2022) discovered that some performance loss is due to undistillable classes and suggested ways to address this.

6.3 Quantization

Mapping high-precision data types to low-precision ones is referred to as *quantization*. Quantization can be applied at different stages in the NLP model-building pipeline to reduce training and inference costs. Various research has shown that low-precision data format can reduce memory consumption by 4x–24x and improve the throughput by 4.5x compared to 32-bit floating point format. Various studies targeted 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 et al., 2020; Dettmers et al., 2022a), ternary (Zhang et al., 2020; Ji et al., 2021; Zadeh et al., 2022), and even binary representations (Bai et al., 2021).

Different components may have different sensitivities regarding their underlying precision, so there is a body of work on mixed-precision quantization. Shen et al. (2020) showed that embedding layers require more precise parameter representations than the attention layer, while Kim et al. (2021) showed that nonlinear functions require more bits than the general matrix multiplication. Others defined quantization as a constrained optimization problem to automatically identify layers where lower precision is sufficient (Hubara et al., 2021). Finally, several studies proposed quantization during training to make them robust against performance loss after quantization (Zafrir et al., 2019; Kim et al., 2021; Stock et al., 2021). For instance, Bai et al. (2021) and Zhang et al. (2020) proposed using knowledge distillation to maintain the accuracy of binarized and ternarized models. These show that component-customized quantization can preserve accuracy while improving efficiency. To maximize the benefit from quantization, one should also consider the available underlying hardware and associated specialized kernels compatible with different bit representations (Noune et al., 2022; Kuzmin et al., 2022).

6.4 Inference Considerations

While efficiency during pre-training and fine-tuning focuses on the computational resources and time required to train and optimize a model, inference efficiency is focused on how well a learned model can perform on new input data in real-world scenarios. Moreover, inference optimization is ultimately context-specific and the requirements vary according to the use-case. Therefore, there is no one-size-fits-all solution to optimizing inference, but instead a plethora of techniques. For instance, while Wu et al. (2022b) combine several methods to achieve utmost model compression, other works improve task-specific mechanisms such as beam-search in MT (Peters and Martins, 2021). Parallelism can also be leveraged to increase inference efficiency, but its effectiveness may depend on the hardware available (Rajbhandari et al., 2022). Dynamic computation techniques, such as early-exit (Schwartz et al., 2020b; Xin et al., 2020) and MoE (Section 3.1), can improve inference efficiency by selectively performing computation only on the parts of the model that are needed for a given input. However, current dynamic computation methods often use eager execution mode, which can prevent them from low-level optimization, as noted by Xu and McAuley (2023). Work focusing on inference efficiency should carefully report the exact target setting (hardware, eager vs. static execution framework). Accordingly, promising directions for optimizing inference efficiency might consider tighter integration across or more general purpose approaches with respect to algorithm, software, and hardware. One recent such example is neural architecture search for hardware-specific efficient transformers (Wang et al., 2020a).

7 Hardware Utilization

Many hardware-specific methods focus on reducing GPU memory consumption, a major bottleneck in transformer models. Others leverage specialized hardware, co-design of hardware, and adaptations targeted to edge devices. Many techniques can be combined and applied across different stages of training and inference (Figure 2) for further efficiency.

7.1 Reducing Optimizer Memory

Optimizers that track gradient history incur a memory cost. Libraries like DeepSpeed (Ren et al., 2021a) allow gradient history to be offloaded from GPU to CPU RAM where computation is performed via efficient AVX instructions. `bitsandbytes` (Dettmers et al., 2022b) uses dynamic block-wise quantization to reduce memory pressure. It splits tensors into blocks and quantizes each block individually. This reduces memory consumption by 75% and improves training times due to reduced inter-GPU communication.

7.2 Specialized Hardware

Specialized NLP hardware has been built using Application Specific Integrated Circuits or Field Programmable Gate Arrays, though it is not yet broadly available. These designs use dedicated units for efficient operations like quantization and pruning (Section 6). For example, Zadeh et al. (2020, 2022), Li et al. (2021), and Qu et al. (2022) support ultra-low-bit and mixed precision computation that cannot be done on CPUs/GPUs; Ham et al. (2020, 2021) and Wang et al. (2021a) design hardware that predicts and prunes redundant heads/tokens and weak attention values in transformers. Qu et al. (2022) present a design that balances the workload to alleviate the irregularity in the pruned attention. Others develop new types of processors and memories optimized for transformer components: Lu et al. (2020) and Liu et al. (2021b) implemented dedicated hardware for softmax and layer normalization respectively, and Tambe et al. (2021) used embedded Resistive RAM—a nonvolatile memory with low latency and energy consumption—to store word embeddings.

7.3 Co-design

Some work optimizes hardware, software, and algorithms jointly, which historically has been a common way to realize efficiency gains (Hooker, 2021). For instance, Lepikhin et al. (2021) demonstrated that improving the underlying compiler can substantially improve parallelization and enable scaling. Other examples for co-design focus on hardware-aware mixture of experts models and attention mechanisms to produce substantial speedups (He et al., 2022a; Rajbhandari et al., 2022; Dao et al., 2022b). Barham et al. (2022) proposed a gang-scheduling approach with parallel asynchronous dispatch that leads to substantial efficiency gains. Finally, Hinton (2022) suggested ‘‘mortal computation’’, an extreme form of co-design, where by training a model that is tailored to a specific hardware, the need to guarantee consistent software behavior across different hardware is reduced, potentially saving computation.

7.4 Edge Devices

Tight compute and memory constraints on edge devices motivate a separate set of efficiency solutions. SqueezeBERT (Iandola et al., 2020) incorporates group convolutions into self-attention to improve efficiency on mobile devices. EdgeFormer (Ge et al., 2022) interleaves self-attention layers with lightweight feed-forward layers and an encoder-heavy parameterization to meet edge memory budgets. GhostBERT (Huang et al., 2021) uses *ghost* modules built on depth-wise separable convolutions used in MobileNets (Howard et al., 2017). LiteTransformer (Wu et al., 2020) uses long-short range attention to encode local context by convolutions for MT in resource-constrained settings. Through quantization `llama.cpp`[1] runs a 7B-parameter LLM on recent mobile phone hardware. Finally, ProFormer (Sankar et al., 2021) reduces runtime and memory via locality sensitive hashing and local projection attention layers.

7.5 Hardware Considerations

To deliver more computational power, vendors pack denser computational units into domain-specific hardware, such as tensor cores in Intel FPGAs, Xilinx AI Engines, and matrix processors in the Google TPU. However, irregularities in the transformer, like sparsity and mixed data types, restrict the use of these resources. We suggest focusing on adapting efficient transformers to existing specialized hardware platforms, including using hardware-optimized data formats like block floating point, and exploring sparsity on dense tensor units.

8 Evaluating Efficiency

Evaluating efficiency requires establishing which computational aspect one aims to minimize. We discuss the two most prominent aspects (FLOP/s and power consumption), and list open challenges.

8.1 Evaluation Measures

Pareto Optimality When improving efficiency, multiple factors often need to be traded off. For instance, longer training time can increase task performance, but simultaneously increase

[1] `https://github.com/ggerganov/llama.cpp`, 20 March 2023.

resource consumption. A principled way to characterize trade-offs is to identify Pareto-optimal solutions (Pareto, 1896), those for which no other system reaches a better or equal task performance with lower resource consumption. As there may be several Pareto-optimal solutions, final choice depends on the application context; a small, average-quality model and a large, higher-quality model can both be optimal. Thus, as long as a model contributes to or extends the Pareto-optimal curve for a given problem and measurement space, it it worthwhile—even if other solutions may use less resources or produce higher quality scores.

Advancing NLP by pushing Pareto barriers is an established practice (Kim et al., 2019; Bogoychev et al., 2020; Behnke and Heafield, 2021). For instance, the WNGT 2020 MT 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 score. Puvis de Chavannes et al. (2021) included power consumption as a trade-off against perplexity to explore Pareto-efficient hyperparameter combinations for transformer models. Finally, Liu et al. (2022b) examined Pareto efficiency for a number of tasks in an attempt to narrow model selection search space.

FLOP/s A frequently reported efficiency measure is the number of floating point operations (FLOPs) and floating points per second (FLOP/s). While these discrete metrics seem 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; and hardware is rarely 100% utilized 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 (Hsu et al., 2005).

Power Consumption There exist various ways 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 for individual computers in a network. Moreover, it does not cover external energy costs such as cooling or networking. Another way is to use 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) introduced a model card for NLP systems that encourages researchers to document efficiency in a consistent manner.

Measuring power consumption programmatically comes with a number of caveats. First, sampling frequency is often restricted 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 energy demand that is missed. Second, inefficiencies such as heat loss are not reported by current APIs and hence do not cover cooling and other system management activities. Third, not all architectures and operating systems are supported. For instance, power consumption under macOS is difficult to manage, and direct figures for TPU power consumption are not available.

Carbon Emissions Carbon emissions are usually computed using the power consumption and the carbon intensity of the marginal energy generation 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; Dodge et al., 2022). For estimating the CO_2 emissions from a specific program execution, APIs such as ElectricityMap[2] 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) recommended using multiple runs for a stable estimate. Furthermore, one needs to consider that zero-carbon program executions still consume energy, and that efficiency does not intrinsically guarantee a reduction in overall resource consumption, as the resulting cost reduction may lead to an increase

[2] https://electricitymap.org.

in demand counteracting any gains, an effect known as Jevons' paradox (Jevons, 1866).

8.2 Open Challenges in Measuring Efficiency

Hardware choice can lead to pronounced differences in certain efficiency measurements such as latency and thoroughput (Lee-Thorp et al., 2022). Properly measuring efficiency remains a major challenge (Cao et al., 2020).

Separating Different Stages It is important to characterize efficiency of pre-training and fine-tuning stages separately (Sections 4 and 5). 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 choices that increase the memory overhead of backward propagation. Further, some optimizers may require substantially more memory than others. Similarly, parameter sharing techniques may show few benefits during training but show memory improvements at inference (Dehghani et al., 2022). Finally, while larger models run more slowly than smaller ones, they converge faster and better compress using methods like pruning and quantization (Li et al., 2020c).

Disagreement Between Cost Factors As partially discussed in Section 7.2, cost indicators may disagree with each other. For instance, MoEs 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, they introduce 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, Chen et al. (2022) and Dao et al. (2022a) found specific sparsity patterns that achieve more predictable speedups with current hardware.

Trade-offs with Other Desiderata One major, but seldom studied, concern when improving efficiency are trade-offs with other desiderata such as fairness and robustness. For instance, Hooker et al. (2020), Renduchintala et al. (2021), and Silva et al. (2021) found that compression techniques such as pruning can amplify existing biases; Mohammadshahi et al. (2022) and Ogueji et al. (2022) further explored these trade-offs in a multilingual setting. So far, only a few studies investigated preserving a model's fairness when increasing its efficiency. To quantify such effects, Xu et al. (2021) proposed a novel metric called loyalty, which measures the resemblance of predicted distributions made by teacher and student models. Hessenthaler et al. (2022) established that many approaches for increasing fairness in NLP models also increase computation, and jointly with work like Wang et al. (2022a) showed that distillation can decrease model fairness. Xu and Hu (2022) studied these effects more systematically, with mixed conclusions. While more positive insights have been found with respect to other desiderata such as out-of-distribution (OOD) generalization (Ahia et al., 2021; Iofinova et al., 2022; Ogueji et al., 2022) and model transfer (Gordon et al., 2020), more work is needed to better understand and benchmark the impact of efficiency beyond accuracy.

9 Model Selection

Finally, we discuss lines of research that opt to efficiently select a well-performing model variant.

9.1 Hyperparameter Search

The performance of machine learning methods can be improved by choosing hyperparameters carefully. 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 inefficient grid search or manual tuning. Complementary approaches are successive halving (SHA; Jamieson and Talwalkar, 2016) and its massively parallel variant, asynchronous SHA (ASHA; Li et al., 2020b), 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.

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., 2022) and *auto-pytorch* (Zimmer et al., 2021). However, with limited computational budgets, both BO and ASHA may fail to identify good settings (Liu and Wang, 2021). It is unclear whether these methods can be used to choose random initial weights or to order training samples, which also affect model performance (Dodge et al., 2020).

9.2 Hyperparameter Transfer

To minimize the number of trials needed to find optimal hyperparameter settings, one can transfer knowledge from other datasets or tasks—similar to how an ML engineer might select reasonable settings by hand. Transferring hyperparameters can be especially beneficial during expensive stages in the NLP pipeline, such as pre-training. 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. Finally, the cost of training can be reduced using μTransfer (Yang et al., 2021), which tunes a small model, then transfers the hyperparameters to a larger model.

9.3 Model Selection Considerations

While identifying an optimal model is crucial in deployment, it raises several challenges around reporting practices (Reimers and Gurevych, 2017; Agarwal et al., 2021) and hyperparameter tuning (Bouthillier and Varoquaux, 2020; Gundersen et al., 2022).[3] A first step towards improved comparability could be to fix the hyperparameter tuning budget (Dodge et al., 2019; Hoffmann et al., 2022), or consider the full search space (Bell et al., 2022).

10 Conclusion

This survey provides a broad overview of considerations for increasing efficiency in modern NLP models, identifying both immediate successes and remaining challenges. Most progress so far has been in model design, typically targeted at a specific computational budget and hardware paradigm. Key challenges include better understanding and modeling trade-offs between end-task performance and resource consumption, and the dependency between hardware choices and software implementations. Furthermore, we note that efficiency in NLP has many definitions and can be achieved in many different ways, but is also subject to various open challenges, and cannot be measured by a single metric. We outline several promising research directions aligned with overcoming these challenges, ranging from approaches that make better use of available data, strategies for reducing the cost of pre-training and fine-tuning large models, to prioritizing the importance of interactions between algorithms, software, and hardware.

Impressive advances in NLP enabled primarily by scaling computation have produced remarkable progress in a short span of time. However, in order to realize the full potential of this technology for a broader swath of society, we must reduce the amount of computation that is required to achieve these remarkable results. We hope that this survey can serve to accelerate advances in this important area of research with great potential for impact both within our field and for society as a whole.

Acknowledgments

This work was initiated at and benefited substantially from the Dagstuhl Seminar 22232: *Efficient and Equitable Natural Language Processing in the Age of Deep Learning*. We further thank Yuki Arase, Jonathan Frankle, Alexander Koller, Alexander Löser, Alexandra Sasha Luccioni, Haritz Puerto, Nils Reimers, Leonardo Riberio, Anna Rogers, Andreas Rücklé, Noah A. Smith, and Thomas Wolf for a fruitful discussion and helpful feedback at the seminar. M.T. and A.M. acknowledge the European Research Council (ERC StG DeepSPIN 758969), EU's Horizon Europe Research and Innovation Actions (UTTER, contract 101070631), and Fundação para a Ciência e Tecnologia through contract UIDB/50008/2020. L.D. acknowledges support of the Independent Research Fund Denmark under project 9131-00131B, Verif-AI, and the Novo Nordisk Foundation project ClinRead, NNF19-OC0059138. Finally, we also thank the TACL reviewers and action editor for helpful discussion and insightful feedback.

[3] For example, when considering compute budget variation when comparing new model development to baselines.

References

Chirag Agarwal, Daniel D'souza, and Sara Hooker. 2022. Estimating example difficulty using variance of gradients. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10368–10378. https://doi.org/10.1109/CVPR52688.2022.01012

Rishabh Agarwal, Max Schwarzer, Pablo Samuel Castro, Aaron C. Courville, and Marc Bellemare. 2021. Deep reinforcement learning at the edge of the statistical precipice. In *Advances in Neural Information Processing Systems*, volume 34, pages 29304–29320. Curran Associates, Inc.

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. https://doi.org/10.18653/v1/2021.emnlp-main.468

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. https://doi.org/10.18653/v1/2021.acl-long.568

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.

Nur Ahmed and Muntasir Wahed. 2020. The de-democratization of AI: Deep learning and the compute divide in artificial intelligence research. *arXiv preprint arXiv:2010.15581v1*.

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. https://doi.org/10.18653/v1/2020.emnlp-main.19

Ahmed Alajrami and Nikolaos Aletras. 2022. How does the pre-training objective affect what large language models learn about linguistic properties? In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 131–147, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-short.16

Uri Alon, Frank Xu, Junxian He, Sudipta Sengupta, Dan Roth, and Graham Neubig. 2022. Neuro-symbolic language modeling with automaton-augmented retrieval. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 468–485. PMLR.

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*.

Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. 2020. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*.

Stephen Bach, Victor Sanh, Zheng Xin Yong, Albert Webson, Colin Raffel, Nihal V. Nayak, Abheesht Sharma, Taewoon Kim, M. Saiful Bari, Thibault Fevry, Zaid Alyafeai, Manan Dey, Andrea Santilli, Zhiqing Sun, Srulik Ben-david, Canwen Xu, Gunjan Chhablani, Han Wang, Jason Fries, Maged Al-shaibani, Shanya Sharma, Urmish Thakker, Khalid Almubarak, Xiangru Tang, Dragomir Radev, Mike Tian-Jian Jiang, and Alexander Rush. 2022. PromptSource: An integrated development environment and repository for natural language prompts. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 93–104, Dublin, Ireland. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.acl-demo.9`

Haoli Bai, Wei Zhang, Lu Hou, Lifeng Shang, Jin Jin, Xin Jiang, Qun Liu, Michael Lyu, and Irwin King. 2021. BinaryBERT: Pushing the limit of BERT quantization. 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 4334–4348, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2021.acl-long.334`

Robert Baldock, Hartmut Maennel, and Behnam Neyshabur. 2021. Deep learning through the lens of example difficulty. In *Advances in Neural Information Processing Systems*, volume 34, pages 10876–10889. Curran Associates, Inc.

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, Daniel Hurt, Michael Isard, Hyeontaek Lim, Ruoming Pang, Sudip Roy, Brennan Saeta, Parker Schuh, Ryan Sepassi, Laurent Shafey, Chandu Thekkath, and Yonghui Wu. 2022. Pathways: Asynchronous distributed dataflow for ML. *Proceedings of Machine Learning and Systems*, 4:430–449.

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.

Samuel Bell, Onno Kampman, Jesse Dodge, and Neil D. Lawrence. 2022. Modeling the machine learning multiverse. In *Advances in Neural Information Processing Systems*.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150v2*.

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. `https://doi.org/10.18653/v1/2022.acl-short.1`

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. In *Proceedings of the Joint Workshop on On-Device Machine Learning & Compact Deep Neural Network Representations, 36th International Conference on Machine Learning*.

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. `https://doi.org/10.18653/v1/2021.findings-emnlp.259`

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.

Xavier Bouthillier and Gaël Varoquaux. 2020. Survey of machine-learning experimental methods at NeurIPS2019 and ICLR2020. Research report, Inria Saclay Ile de France.

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, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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.

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75. `https://doi.org/10.1023/A:1007379606734`

Beidi Chen, Tri Dao, Kaizhao Liang, Jiaming Yang, Zhao Song, Atri Rudra, and Christopher Re. 2022. Pixelated butterfly: Simple and efficient sparse training for neural network models. In *International Conference on Learning Representations*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374v2*.

Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. 2019. Generating long sequences with sparse transformers. *arXiv preprint arXiv:1904.10509v1*.

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:2204.02311v5.

Kevin Clark, Minh-Thang Luong, Quoc V. Le, and Christopher D. Manning. 2020. ELECTRA: Pre-training text encoders as discriminators rather than generators. In *International Conference on Learning Representations*.

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.

Corinna Cortes, Mehryar Mohri, Michael Riley, and Afshin Rostamizadeh. 2008. Sample selection bias correction theory. In *Algorithmic Learning Theory*, pages 38–53, Berlin, Heidelberg. Springer Berlin Heidelberg.

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, Nimit S. Sohoni, Arjun Desai, Michael Poli, Jessica Grogan, Alexander Liu, Aniruddh Rao, Atri Rudra, and Christopher Ré. 2022a. Monarch: Expressive structured matrices for efficient and accurate training. In *International Conference on Machine Learning*, pages 4690–4721. PMLR.

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

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

Mostafa Dehghani, Yi Tay, Anurag Arnab, Lucas Beyer, and Ashish Vaswani. 2022. The efficiency misnomer. In *International Conference on Learning Representations*.

Leon Derczynski. 2020. Power consumption variation over activation functions. *arXiv preprint arXiv:2006.07237v1*.

Tim Dettmers, Mike Lewis, Younes Belkada, and Luke Zettlemoyer. 2022a. GPT3.int8(): 8-bit matrix multiplication for transformers at scale. In *Advances in Neural Information Processing Systems*.

Tim Dettmers, Mike Lewis, Sam Shleifer, and Luke Zettlemoyer. 2022b. 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, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A. Smith. 2019. Show your work: Improved reporting of experimental results. 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 2185–2194, Hong Kong, China. Association for Computational Linguistics. `https://doi.org/10.18653/v1/D19-1224`

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pre-trained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305v1*.

Jesse Dodge, Taylor Prewitt, Remi Tachet des Combes, Erika Odmark, Roy Schwartz, Emma Strubell, Alexandra Sasha Luccioni, Noah A. Smith, Nicole DeCario, and Will Buchanan. 2022. Measuring the carbon intensity of AI in cloud instances. In *2022 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '22, pages 1877–1894, New York, NY, USA. Association for Computing Machinery. `https://doi.org/10.1145/3531146.3533234`

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.

Daniel D'souza, Zach Nussbaum, Chirag Agarwal, and Sara Hooker. 2021. A tale of two long tails. *arXiv preprint arXiv:2107.13098v1*.

Nan Du, Yanping Huang, Andrew M. Dai, Simon Tong, Dmitry Lepikhin, Yuanzhong Xu, Maxim Krikun, Yanqi Zhou, Adams Wei Yu, Orhan Firat, Barret Zoph, Liam Fedus, Maarten P Bosma, Zongwei Zhou, Tao Wang, Emma Wang, Kellie Webster, Marie Pellat, Kevin Robinson, Kathleen Meier-Hellstern, Toju Duke, Lucas Dixon, Kun Zhang, Quoc Le, Yonghui Wu, Zhifeng Chen, and Claire Cui. 2022. GLaM: Efficient scaling of language models with mixture-of-experts. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 5547–5569. PMLR.

Yann Dubois, Gautier Dagan, Dieuwke Hupkes, and Elia Bruni. 2020. Location attention for extrapolation to longer sequences. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 403–413, Online. Association for Computational Linguistics.

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. https://doi.org/10.18653/v1/2020.emnlp-main.638

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. https://doi.org/10.1016/0010-0277(93)90058-4, PubMed: 8403835

Kawin Ethayarajh, Yejin Choi, and Swabha Swayamdipta. 2022. Understanding dataset difficulty with $\mathcal{V}$-usable information. In *International Conference on Machine Learning*, pages 5988–6008. PMLR.

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

William Fedus, Jeff Dean, and Barret Zoph. 2022a. A review of sparse expert models in deep learning. *arXiv preprint arXiv:2209.01667v1*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022b. 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. 2022. Auto-Sklearn 2.0: Hands-free autoML via meta-learning. *Journal of Machine Learning Research*, 23(261):1–61.

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:1902.09574v1*.

Tao Ge, Si-Qing Chen, and Furu Wei. 2022. EdgeFormer: A parameter-efficient transformer for on-device seq2seq generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10786–10798, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Timnit Gebru, Jamie Morgenstern, Briana Vecchione, Jennifer Wortman Vaughan, Hanna Wallach, Hal Daumé Iii, and Kate Crawford. 2021. Datasheets for datasets. *Communications of the ACM*, 64(12):86–92. https://doi.org/10.1145/3458723

Daniel Gissin and Shai Shalev-Shwartz. 2019. Discriminative active learning. *arXiv preprint arXiv:1907.06347v1*.

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. https://doi.org/10.18653/v1/2020.repl4nlp-1.18

Jianping Gou, Baosheng Yu, Stephen J. Maybank, and Dacheng Tao. 2021. Knowledge distillation: A survey. *International Journal of Computer Vision*, 129(6):1789–1819. https://doi.org/10.1007/s11263-021-01453-z

Albert Gu, Karan Goel, Ankit Gupta, and Christopher Ré. 2022a. On the parameterization and initialization of diagonal state space models. In *Advances in Neural Information Processing Systems*.

Albert Gu, Karan Goel, and Christopher Re. 2022b. 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*.

Odd Erik Gundersen, Kevin Coakley, and Christine Kirkpatrick. 2022. Sources of irreproducibility in machine learning: A review. *arXiv preprint arXiv:2204.07610v1*.

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. `https://doi.org/10.18653/v1/2021.acl-long.378`

Ankit Gupta, Albert Gu, and Jonathan Berant. 2022. Diagonal state spaces are as effective as structured state spaces. In *Advances in Neural Information Processing Systems*.

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.

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.

Song Han, Jeff Pool, John Tran, and William Dally. 2015. Learning both weights and connections for efficient neural networks. *Advances in Neural Information Processing Systems*, 28.

Michael Hassid, Hao Peng, Daniel Rotem, Jungo Kasai, Ivan Montero, Noah A. Smith, and Roy Schwartz. 2022. How much does attention actually attend? Questioning the importance of attention in pre-trained transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 1403–1416, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jiaao He, Jidong Zhai, Tiago Antunes, Haojie Wang, Fuwen Luo, Shangfeng Shi, and Qin Li. 2022a. 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, pages 120–134, New York, NY, USA. Association for Computing Machinery.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022b. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Kaiming He, Ross Girshick, and Piotr Dollár. 2019. Rethinking ImageNet pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. `https://doi.org/10.1109/ICCV.2019.00502`

Pengcheng He, Jianfeng Gao, and Weizhu Chen. 2023. DeBERTaV3: Improving DeBERTa using electra-style pre-training with gradient-disentangled embedding sharing. In *The Eleventh International Conference on Learning Representations*.

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 Bingler, and Markus Leippold. 2022. Towards climate awareness in NLP research. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2480–2494, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Marius Hessenthaler, Emma Strubell, Dirk Hovy, and Anne Lauscher. 2022. Bridging fairness and environmental sustainability in natural language processing. In *Proceedings of the 2022*

Conference on Empirical Methods in Natural Language Processing, pages 7817–7836, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Geoffrey Hinton. 2022. The forward-forward algorithm: Some preliminary investigations. *arXiv preprint arXiv:2212.13345v1*.

Geoffrey Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. In *NeurIPS Deep Learning and Representation Learning Workshop*.

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.

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

Sara Hooker. 2021. The hardware lottery. *Communications of the ACM*, 64:58–65. https://doi.org/10.1145/3467017

Sara Hooker, Nyalleng Moorosi, Gregory Clark, Samy Bengio, and Emily Denton. 2020. Characterising bias in compressed models. *arXiv preprint arXiv:2010.03058v1*.

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 preprint arXiv:1704.04861v1*.

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. https://doi.org/10.18653/v1/2021.acl-long.509

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. https://doi.org/10.18653/v1/2020.sustainlp-1.17

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. https://doi.org/10.1109/CVPR52688.2022.01195

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. https://doi.org

/10.1162/neco.1991.3.1.79, PubMed: 31141872

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. https://doi.org/10.18653/v1/2020.findings-emnlp.372

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361v1*.

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. https://doi.org/10.18653/v1/2021.acl-long.564

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.

Rabeeh Karimi Mahabadi, Luke Zettlemoyer, James Henderson, Lambert Mathias, Marzieh Saeidi, Veselin Stoyanov, and Majid Yazdani. 2022. Prompt-free and efficient few-shot learning with language models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3638–3652, Dublin, Ireland. Association for Computational Linguistics. https://doi.org/10.18653/v1/2022.acl-long.254

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 *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Sehoon Kim, Amir Gholami, Zhewei Yao, Michael W. Mahoney, and Kurt Keutzer. 2021. I-BERT: Integer-only BERT quantization. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 5506–5518. PMLR.

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. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

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

Jan-Christoph Klie, Richard Eckart de Castilho, and Iryna Gurevych. 2020. From zero to hero: Human-in-the-loop entity linking in low resource domains. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6982–6993, Online. 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. `https://doi.org/10.1162/tacl_a_00447`

M. Kumar, Benjamin Packer, and Daphne Koller. 2010. Self-paced learning for latent variable models. In *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc.

Andrey Kuzmin, Mart Van Baalen, Yuwei Ren, Markus Nagel, Jorn Peters, and Tijmen Blankevoort. 2022. FP8 quantization: The power of the exponent. In *Advances in Neural Information Processing Systems*.

Imad Lakim, Ebtesam Almazrouei, Ibrahim Abualhaol, Merouane Debbah, and Julien Launay. 2022. A holistic assessment of the carbon footprint of Noor, a very large Arabic language model. In *Proceedings of BigScience Episode #5 – Workshop on Challenges & Perspectives in Creating Large Language Models*, pages 84–94, virtual+Dublin. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.bigscience-1.8`

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*.

Ronan Le Bras, Swabha Swayamdipta, Chandra Bhagavatula, Rowan Zellers, Matthew Peters, Ashish Sabharwal, and Yejin Choi. 2020. Adversarial filters of dataset biases. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1078–1088. PMLR.

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. `https://doi.org/10.1162/coli_a_00436`

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. `https://doi.org/10.18653/v1/2022.naacl-main.319`

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*.

Jure Leskovec, Anand Rajaraman, and Jeffrey David Ullman. 2020. *Mining of Massive Data Sets*. Cambridge University Press.

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, London. Springer London.

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. `https://doi.org/10.18653/v1/2020.acl-main.703`

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020b. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Belinda Z. Li, Gabriel Stanovsky, and Luke Zettlemoyer. 2020a. Active learning for coreference resolution using discrete annotation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8320–8331, Online. Association for Computational Linguistics.

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

Huayang Li, Yixuan Su, Deng Cai, Yan Wang, and Lemao Liu. 2022a. A survey on retrieval-augmented text generation. *arXiv preprint arXiv:2202.01110v1*.

Liam Li, Kevin Jamieson, Afshin Rostamizadeh, Ekaterina Gonina, Jonathan Ben-tzur, Moritz Hardt, Benjamin Recht, and Ameet Talwalkar. 2020b. A system for massively parallel hyperparameter tuning. In *Third Conference on Systems and Machine Learning*.

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. `https://doi.org/10.1109/TPDS.2020.3047371`

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.

Yuhong Li, Tianle Cai, Yi Zhang, Deming Chen, and Debadeepta Dey. 2022b. What makes convolutional models great on long sequence modeling? *arXiv preprint arXiv:2210.09298v1*.

Zhuohan Li, Eric Wallace, Sheng Shen, Kevin Lin, Kurt Keutzer, Dan Klein, and Joey Gonzalez. 2020c. Train big, then compress: Rethinking model size for efficient training and inference of transformers. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 5958–5968. PMLR.

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, Muqeeth Mohammed, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022a. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. In *Advances in Neural Information Processing Systems*.

Ming Liu, Wray Buntine, and Gholamreza Haffari. 2018. Learning to actively learn neural machine translation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 334–344, Brussels, Belgium. Association for Computational Linguistics.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Computing Surveys*, 55(9). https://doi.org/10.1145/3560815

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020. Fast-BERT: 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.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2021a. GPT understands, too. *arXiv preprint arXiv:2103.10385v1*.

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.

Zejian Liu, Gang Li, and Jian Cheng. 2021b. Hardware acceleration of fully quantized BERT for efficient natural language processing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE)*.

Christos Louizos, Max Welling, and Diederik P. Kingma. 2018. Learning sparse neural networks through L_0 regularization. In *International Conference on Learning Representations*.

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*.

Xuezhe Ma, Chunting Zhou, Xiang Kong, Junxian He, Liangke Gui, Graham Neubig, Jonathan May, and Luke Zettlemoyer. 2023. Mega: Moving average equipped gated attention. In *The Eleventh International Conference on Learning Representations*.

Francesca Manes-Rossi, Adriana Tiron-Tudor, Giuseppe Nicolò, and Gianluca Zanellato. 2018. Ensuring more sustainable reporting in europe using non-financial disclosure—De facto and de jure evidence. *Sustainability*, 10(4):1162. `https://doi.org/10.3390/su10041162`

Katerina Margatina, Giorgos Vernikos, Loïc Barrault, and Nikolaos Aletras. 2021. Active learning by acquiring contrastive examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 650–663, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Pedro Martins, Zita Marinho, and Andre Martins. 2022a. Efficient machine translation domain adaptation. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 23–29, Dublin, Ireland and Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.spanlp-1.3`

Pedro Henrique Martins, Zita Marinho, and Andre Martins. 2022b. ∞-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.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022c. Chunk-based nearest neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4228–4245, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Harsh Mehta, Ankit Gupta, Ashok Cutkosky, and Behnam Neyshabur. 2023. Long range language modeling via gated state spaces. In *The Eleventh International Conference on Learning Representations*.

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. `https://doi.org/10.18653/v1/2022.findings-acl.47`

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.

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. `https://doi.org/10.18653/v1/2020.sustainlp-1.23`

Alireza Mohammadshahi, Vassilina Nikoulina, Alexandre Berard, Caroline Brun, James Henderson, and Laurent Besacier. 2022. What do compressed multilingual machine translation models forget? In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 4308–4329, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nafise Moosavi, Quentin Delfosse, Kristian Kersting, and Iryna Gurevych. 2022. Adaptable adapters. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3742–3753, Seattle, United States. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2022.naacl-main.274`

Hesham Mostafa and Xin Wang. 2019. Parameter efficient training of deep convolutional neural networks by dynamic sparse reparameterization. In *Proceedings of the 36th International Conference on Machine Learning*, pages 4646–4655. PMLR.

Basil Mustafa, Carlos Riquelme Ruiz, Joan Puigcerver, Rodolphe Jenatton, and Neil Houlsby. 2022. Multimodal contrastive learning with LIMoE: The language-image mixture of experts. In *Advances in Neural Information Processing Systems*.

Behnam Neyshabur, Hanie Sedghi, and Chiyuan Zhang. 2020. What is being transferred in transfer learning? In *Advances in Neural Information Processing Systems*, volume 33, pages 512–523. Curran Associates, Inc.

Badreddine Noune, Philip Jones, Daniel Justus, Dominic Masters, and Carlo Luschi. 2022. 8-bit numerical formats for deep neural networks. *arXiv preprint arXiv:2206.02915v1*.

Kelechi Ogueji, Orevaoghene Ahia, Gbemileke Onilude, Sebastian Gehrmann, Sara Hooker, and Julia Kreutzer. 2022. Intriguing properties of compression on multilingual models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9092–9110, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Vilfredo Pareto. 1896. *Cours d'Économie Politique professé à l'Université de Lausanne*, volume 1. F. Rouge.

David Patterson, Joseph Gonzalez, Quoc Le, Chen Liang, Lluis-Miquel Munguia, Daniel Rothchild, David So, Maud Texier, and Jeff Dean. 2021. Carbon emissions and large neural network training. *arXiv preprint arXiv: 2104.10350v3*.

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. `https://doi.org/10.18653/v1/2021.naacl-main.210`

Ben Peters, Vlad Niculae, and André F. T. Martins. 2019. Sparse sequence-to-sequence models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1504–1519, Florence, Italy. 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. `https://doi.org/10.18653/v1/N18-1202`

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. `https://doi.org/10.18653/v1/D19-1250`

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. AdapterHub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 46–54, Online. Association for Computational Linguistics. `https://doi.org/10.18653/v1/2020.emnlp-demos.7`

Edoardo M. Ponti, Alessandro Sordoni, and Siva Reddy. 2022. Combining modular skills in multitask learning. *arXiv preprint arXiv: 2202.13914v1*.

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.

https://doi.org/10.18653/v1/2020.findings-emnlp.1

Ofir Press, Noah Smith, and Mike Lewis. 2022. Train short, test long: Attention with linear biases enables input length extrapolation. In *International Conference on Learning Representations*.

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. https://doi.org/10.18653/v1/2021.acl-long.427

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, Virtual. Association for Computational Linguistics.

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. https://doi.org/10.18653/v1/N18-3014

Alec Radford, Rafal Jozefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *arXiv preprint arXiv:1704.01444v2*.

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

Jack W. Rae, Sebastian Borgeaud, Trevor Cai, Katie Millican, Jordan Hoffmann, H. Francis Song, John Aslanides, Sarah Henderson, Roman Ring, Susannah Young, Eliza Rutherford, Tom Hennigan, Jacob Menick, Albin Cassirer, Richard Powell, George van den Driessche, Lisa Anne Hendricks, Maribeth Rauh, Po-Sen Huang, Amelia Glaese, Johannes Welbl, Sumanth Dathathri, Saffron Huang, Jonathan Uesato, John Mellor, Irina Higgins, Antonia Creswell, Nat McAleese, Amy Wu, Erich Elsen, Siddhant M. Jayakumar, Elena Buchatskaya, David Budden, Esme Sutherland, Karen Simonyan, Michela Paganini, Laurent Sifre, Lena Martens, Xiang Lorraine Li, Adhiguna Kuncoro, Aida Nematzadeh, Elena Gribovskaya, Domenic Donato, Angeliki Lazaridou, Arthur Mensch, Jean-Baptiste Lespiau, Maria Tsimpoukelli, Nikolai Grigorev, Doug Fritz, Thibault Sottiaux, Mantas Pajarskas, Toby Pohlen, Zhitao Gong, Daniel Toyama, Cyprien de Masson d'Autume, Yujia Li, Tayfun Terzi, Vladimir Mikulik, Igor Babuschkin, Aidan Clark, Diego de Las Casas, Aurelia Guy, Chris Jones, James Bradbury, Matthew J. Johnson, Blake A. Hechtman, Laura Weidinger, Iason Gabriel, William S. Isaac, Edward Lockhart, Simon Osindero, Laura Rimell, Chris Dyer, Oriol Vinyals, Kareem Ayoub, Jeff Stanway, Lorrayne Bennett, Demis Hassabis, Koray Kavukcuoglu, and Geoffrey Irving. 2021. Scaling language models: Methods, analysis & insights from training gopher. *arXiv preprint arXiv:2112.11446v2*.

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. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 18332–18346. PMLR.

Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. 2017. Learning multiple visual domains with residual adapters. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

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. Association for Computational Linguistics, Punta Cana, Dominican Republic. `https://doi.org/10.18653/v1/2021.findings-emnlp.344`

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics. `https://doi.org/10.18653/v1/D17-1035`

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. USENIX Association.

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 Computing Surveys*, 54(9). `https://doi.org/10.1145/3472291`

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. `https://doi.org/10.18653/v1/2021.acl-short.15`

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. `https://doi.org/10.1162/tacl_a_00353`

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. `https://doi.org/10.18653/v1/2021.emnlp-main.626`

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098v1*.

Hassan Sajjad, Fahim Dalvi, Nadir Durrani, and Preslav Nakov. 2023. On the effect of dropping layers of pre-trained transformer models. *Computer Speech & Language*, 77:101429. `https://doi.org/10.1016/j.csl.2022.101429`

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. In *NeurIPS EMC^2 Workshop*.

Victor Sanh, Albert Webson, Colin Raffel, Stephen Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Arun Raja, Manan Dey, M. Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Teven Le Scao, Stella Biderman, Leo Gao, Thomas Wolf, and

Alexander M. Rush. 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. https://doi.org/10.18653/v1/2021.eacl-main.246

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. https://doi.org/10.18653/v1/2021.naacl-main.185

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2020a. Green AI. *Communications of the ACM (CACM)*, 63(12):54–63. https://doi.org/10.1145/3381831

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. https://doi.org/10.18653/v1/2020.acl-main.593

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)*.

Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. 2018. Self-attention with relative position representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 464–468, New Orleans, Louisiana. Association for Computational Linguistics. https://doi.org/10.18653/v1/N18-2074

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(5):8815–8821. Number: 5. https://doi.org/10.1609/aaai.v34i05.6409

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. https://doi.org/10.18653/v1/2020.emnlp-main.346

Shoaib Ahmed Siddiqui, Nitarshan Rajkumar, Tegan Maharaj, David Krueger, and Sara Hooker. 2021. Metadata archaeology: Unearthing data subsets by leveraging training dynamics. *arXiv preprint arXiv:2209.10015v1*.

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. https://doi.org/10.18653/v1/2021.naacl-main.189

Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. 2012. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A. Alemi, and Andrew G. Wilson. 2021. Does knowledge distillation really work? In *Advances in Neural Information Processing Systems*, volume 34, pages 6906–6919. Curran Associates, Inc.

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. https://doi.org/10.18653/v1/P19-1355

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 A. Raffel. 2021. Training neural networks with fixed sparse masks. In *Advances in Neural Information Processing Systems*, volume 34, pages 24193–24205. Curran Associates, Inc.

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. https://doi.org/10.18653/v1/2020.emnlp-main.746

Thierry Tambe, Coleman Hooper, Lillian Pentecost, Tianyu Jia, En-Yu Yang, Marco Donato, Victor Sanh, Paul Whatmough, Alexander M. Rush, David Brooks, and Gu-Yeon Wei. 2021. EdgeBERT: Sentence-level energy optimizations for latency-aware multi-task NLP inference. In *MICRO-54: 54th Annual IEEE/ACM International Symposium on Microarchitecture*, MICRO '21, pages 830–844, New York, NY, USA. Association for Computing Machinery. https://doi.org/10.1145/3466752.3480095

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, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. 2021. Long range arena : A benchmark for efficient transformers. In *International Conference on Learning Representations*.

Yi Tay, Mostafa Dehghani, Dara Bahri, and Donald Metzler. 2022. Efficient transformers: A survey. *ACM Computing Surveys*. https://doi.org/10.1145/3530811

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:2102.01670v2*.

Neil C. Thompson, Kristjan Greenewald, Keeheon Lee, and Gabriel F. Manso. 2020. The computational limits of deep learning. *arXiv preprint arXiv:2007.05558v1*.

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.

Mojtaba Valipour, Mehdi Rezagholizadeh, Ivan Kobyzev, and Ali Ghodsi. 2022. DyLoRA: Parameter efficient tuning of pre-trained models using dynamic search-free low rank adaptation. In *2nd Workshop on Efficiennt Natural Language and Speech Processing, (NeurIPS workshops)*, pages 1–6.

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. https://doi.org/10.18653/v1/P19-1580

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, Zhanghao Wu, Zhijian Liu, Han Cai, Ligeng Zhu, Chuang Gan, and Song Han. 2020a. HAT: Hardware-aware transformers for efficient natural language processing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7675–7688, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.acl-main.686

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. https://doi.org/10.1109/HPCA51647.2021.00018

Serena Wang, Harikrishna Narasimhan, Yichen Zhou, Sara Hooker, Michal Lukasik, and Aditya Krishna Menon. 2022a. Robust distillation for worst-class performance. *arXiv preprint arXiv:2206.06479v1*.

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:2112.08152v1*.

Yaqing Wang, Sahaj Agarwal, Subhabrata Mukherjee, Xiaodong Liu, Jing Gao, Ahmed Hassan Awadallah, and Jianfeng Gao. 2022b. AdaMix: Mixture-of-adaptations for parameter-efficient model tuning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5744–5760, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Ziheng Wang, Jeremy Wohlwend, and Tao Lei. 2020b. Structured pruning of large language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6151–6162, Online. Association for Computational Linguistics.

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. *Transactions on Machine Learning Research*. Survey Certification.

Ying Wei, Peilin Zhao, and Junzhou Huang. 2021. Meta-learning hyperparameter performance prediction with neural processes. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 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:2202.08005v1*.

Carole-Jean Wu, Ramya Raghavendra, Udit Gupta, Bilge Acun, Newsha Ardalani, Kiwan Maeng, Gloria Chang, Fiona Aga, Jinshi Huang, Charles Bai, Michael Gschwind, Anurag Gupta, Myle Ott, Anastasia Melnikov, Salvatore Candido, David Brooks, Geeta Chauhan, Benjamin Lee, Hsien-Hsin Lee, Bugra Akyildiz, Maximilian Balandat, Joe Spisak, Ravi Jain, Mike Rabbat, and Kim Hazelwood. 2022a. Sustainable AI: Environmental implications, challenges and opportunities. In *Proceedings of Machine Learning and Systems*, volume 4, pages 795–813.

Xiaoxia Wu, Zhewei Yao, Minjia Zhang, Conglong Li, and Yuxiong He. 2022b. Extreme compression for pre-trained transformers made simple and efficient. In *Advances in Neural Information Processing Systems*.

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*.

Mengzhou Xia, Zexuan Zhong, and Danqi Chen. 2022. Structured pruning learns compact and accurate models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1513–1528, Dublin, Ireland. Association for Computational Linguistics.

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. `https://doi.org/10.18653/v1/2020.acl-main.204`

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 and Julian McAuley. 2023. A survey on dynamic neural networks for natural language processing. In *Findings of EACL*.

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:2201.08542v1*.

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. In *Advances in Neural Information Processing Systems*, volume 34, pages 17084–17097. Curran Associates, Inc.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373. `https://doi.org/10.1162/tacl_a_00371`

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. `https://doi.org/10.18653/v1/2022.acl-long.519`

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. https://doi.org/10.1109/MICRO50266.2020.00071

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. https://doi.org/10.1145/3470496.3527438

Ofir Zafrir, Guy Boudoukh, Peter Izsak, and Moshe Wasserblat. 2019. Q8BERT: Quantized 8bit BERT. In *2019 Fifth Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition (EMC^2-NIPS)*, pages 36–39. https://doi.org/10.1109/EMC2-NIPS53020.2019.00016

Ofir Zafrir, Ariel Larey, Guy Boudoukh, Haihao Shen, and Moshe Wasserblat. 2021. Prune once for all: Sparse pre-trained language models. *arXiv preprint arXiv:2111.05754v1*.

Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, and Amr Ahmed. 2020. Big bird: Transformers for longer sequences. In *Advances in Neural Information Processing Systems*, volume 33, pages 17283–17297. Curran Associates, Inc.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics. https://doi.org/10.18653/v1/D18-1009

Shuangfei Zhai, Walter Talbott, Nitish Srivastava, Chen Huang, Hanlin Goh, Ruixiang Zhang, and Josh Susskind. 2021. An attention free transformer. *arXiv preprint arXiv:2105.14103v1*.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, Todor Mihaylov, Myle Ott, Sam Shleifer, Kurt Shuster, Daniel Simig, Punit Singh Koura, Anjali Sridhar, Tianlu Wang, and Luke Zettlemoyer. 2022. OPT: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068v4*.

Wei Zhang, Lu Hou, Yichun Yin, Lifeng Shang, Xiao Chen, Xin Jiang, and Qun Liu. 2020. TernaryBERT: Distillation-aware ultra-low bit BERT. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 509–521, Online. Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.emnlp-main.37

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*, 8393–408.

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. https://doi.org/10.18653/v1/N19-1189

Mingjun Zhao, Haijiang Wu, Di Niu, and Xiaoli Wang. 2020. Reinforced curriculum learning on pre-trained 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.

Yichen Zhu, Ning Liu, Zhiyuan Xu, Xin Liu, Weibin Meng, Louis Wang, Zhicai Ou, and Jian Tang. 2022. Teach less, learn more: On the undistillable classes in knowledge distillation. In *Advances in Neural Information Processing Systems*.

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. https://doi.org/10.1109/TPAMI.2021.3067763, PubMed: 33750687

Barret Zoph, Irwan Bello, Sameer Kumar, Nan Du, Yanping Huang, Jeff Dean, Noam Shazeer, and William Fedus. 2022. Designing effective sparse expert models. In *2022 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1044–1044. IEEE Computer Society. https://doi.org/10.1109/IPDPSW55747.2022.00171