Towards Accurate and Reliable Energy Measurement of NLP Models

Qingqing Cao, Aruna Balasubramanian, Niranjan Balasubramanian
Department of Computer Science
Stony Brook University
Stony Brook, NY 11794, USA
{qicao,arunab,niranjan}@cs.stonybrook.edu

Abstract

Accurate and reliable measurement of energy consumption is critical for making well-informed design choices when choosing and training large scale NLP models. In this work, we show that existing software-based energy measurements are not accurate because they do not take into account hardware differences and how resource utilization affects energy consumption. We conduct energy measurement experiments with four different models for a question answering task. We quantify the error of existing software based energy measurements by using a hardware power meter that provides highly accurate energy measurements. Our key takeaway is the need for a more accurate energy estimation model that takes into account hardware variabilities and the non-linear relationship between resource utilization and energy consumption. We release the code and data at https://github.com/csarron/sustainlp2020-energy.

1 Introduction

State-of-the-art NLP models of today (Devlin et al., 2019; Liu et al., 2019; Raffel et al., 2020) consume large amounts of energy. Such high-levels of energy consumption adds to the worsening global warming and can cause significant social health and safety impacts (Glo; Rolnick et al., 2019). Recent studies have raised awareness of the carbon footprints and potential energy impacts and suggest ways to estimate and reduce consumption (Strubell et al., 2019; Schwartz et al., 2019).

The success of these and future efforts depend on our ability to accurately and reliably estimate the energy consumption of NLP models. A common technique to predict the energy consumption is to measure the utilization of hardware components involved in the computation—the CPU, the GPU, and memory. Each of these components is associated with a single power counter value that is provided by the underlying hardware; this power counter represents the power drawn of a given component. The total energy consumption is computed as the sum of the (utilization $\times$ power counter) of the CPU, GPU, and memory, which is then adjusted by a compensation constant (Henderson et al., 2020; Strubell et al., 2019). We call this technique software-based power measurement.

However there are two potential sources of inaccuracies in the software-based power measurement techniques. First, the software tools are known to be inaccurate because they only consider the energy consumed by three specific hardware components, which may not reflect the energy consumption of the entire system. Second, accurately mapping hardware utilization to the energy consumption is a difficult problem. The mapping depends on the underlying hardware make and type, energy is not always linearly related to the utilization (Pathak et al., 2011, 2012), and energy consumption often continues even after the NLP model has finished running (Burtscher et al., 2014).

In this work, we use a hardware power meter to measure ground truth energy consumption, which is more accurate. Our goal is to quantify how far software-based measurements are from the hardware energy measurements. We compare the energy estimates obtained using prior software based models for four Transformer-based NLP models fine-tuned for a question answering (QA) task.

In the experiments, we find that (1) software energy estimates can differ from the hardware power measurements by 20% on average. Further, the standard deviations are 2$\times$ larger than hardware power meters. (2) Power-models need to take into account the underlying hardware, make, and configuration. Hardware-agnostic energy measurements results in large errors, for example, when applied to machines with different configurations (e.g. dif-

Proceedings of SustaiNLP: Workshop on Simple and Efficient Natural Language Processing, pages 141–148
Online, November 20, 2020.

ferent GPU models, # of GPUs used).

Finally, we show the importance of accurate power-models to make the right accuracy/energy trade-off. Ground-truth energy measurements using a hardware meter show that RoBERTa-base incurs 13% more energy on average. But RoBERTa-base can answer 2.2% more questions correctly over BERT-base. However, existing power-models estimate the additional power consumption of RoBERTa-base to be 25%. Such inaccuracies can lead to wrong conclusions and poor optimizations for model practitioners. The results in this paper suggests that we need better estimation models that are calibrated to account for hardware variabilities and the non-linear relationship between power consumption and resource utilization.

2 Experiments Methodology

In this section, we describe our setup and methodology for energy measurements. We focus on energy consumption of inference for a QA task using a hardware power meter. For comparison purposes, we track software reported energy values as well.

2.1 Setup

Devices: We use 2 GPU-equipped desktop PCs as the target hardware for running our models. See Table 1 for details.

We fine-tune and perform inference in all 4 models on the SQuAD v1.1 question answering dataset (Rajpurkar et al., 2016) using PyTorch (Paszke et al., 2019) v1.6 through the HuggingFace Transformers (Wolf et al., 2020) library. The four models we study are — BERT-base (Devlin et al., 2019), RoBERTa-base (Liu et al., 2019), MobileBERT (Sun et al., 2020), and DistillBERT (Sanh et al., 2020).

Specification	**PC1**	**PC2**
CPU	Intel i9-7900X	Intel i7-6800K
Memory	32 GiB	32 GiB
GPU	2× GTX 1080 Ti	2× GTX 1070
GPU Memory	11.2 GiB per GPU	8 GiB per GPU
Storage	1 TiB SSD	1 TiB SSD

Table 1: Target hardware specifications.

Hardware-based Measurements We use the WattsUP power meter (Wat)[1] to measure *all* of energy consumed by a PC. The WattsUpMeter is used to power the computer, and the power meter records the passthrough current and voltage values every 1 second. This allows us to accurately measure the power draw at a 1 second granularity. Figure 1 shows the energy measurement setup. We obtain current, voltage, and timestamp values from the power meter's built-in USB port. The energy (e) consumed during a time period is then calculated using the sampled current (I_t) and voltage (V_t) values in that period: $e = \sum_t V_t I_t$.

Software-based Measurements: For comparisons, we use the software-based energy measurements provided by the *experiment-impact-tracker* framework (Henderson et al., 2020) which estimates energy as a function of the GPU, CPU, and memory utilization. More details about the model can be found in §3.2.

2.2 Methodology

For each NLP model, we obtain the energy measurements over a random sample of 1000 questions from the SQuAD 1.1 dev split. We repeat these measurements over 10 runs and report the average and standard deviation of energy values. We use 1 GPU to run all experiments, but show the energy measurements accuracy for multiple GPUs in §3.2. Since it is common to batch process inputs on GPUs, we benchmark batch size 1 and batch sizes from 2 to 16 with step 2 [2].

To guarantee the consistency and reliability of the hardware energy measurement, we cool down the PCs after each experiment finishes to avoid potential overheating issue that can cause subsequent energy distortions. We measure the standby power consumption (when the CPU load is $< 0.1\%$) and ensure before running the experiments that the PC does not draw more than the standby power. Further, no other application is running during our experiments.

We record the start and end timestamp of the benchmarked program, and extract the energy values by comparing and aligning the timestamps from the power meter logs. All the energy and latency numbers are end to end, except in §3.4 where we extract the numbers for the prediction part only. In §3.4, we study the latency speedups for model prediction, whereas the latency numbers for data

[1]The device is available on Amazon https://amzn.to/2EoP0tU

[2]We tried larger batch sizes, but found the energy and latency values to be similar for batch sizes between 18 and 32, therefore, we omit numbers with batch size larger than 16 for brevity.

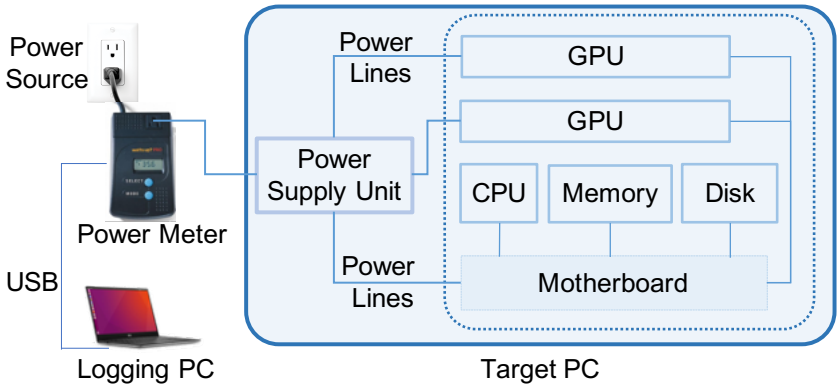


Figure 1: Illustration of the energy measurement setup using a hardware power meter.

loading, startup and cleanup are often comparable to model inference given that we only run for 1000 questions. In the real case, the startup and cleanup costs will be amortized if running millions or billions of the model inference.

3 Energy Results of NLP Models

In this section, we discuss the energy results of the four Transformer-based NLP models on a question answering task.

3.1 Existing Software-Based Energy Measurements Are Not Accurate

We use the energy values recorded by the hardware power meter as ground truth, and report both error percentage (|true energy - software-based energy measurement|/true energy) and standard deviations of the software energy measurements for all four NLP models.

Figure 2a shows that the error of the software measurements ranges from 2% to as much as 47%. In more than 90% of the runs the error is at least 20%, and for a fifth of the runs the error is at least 30%. On average the error percentages are substantial for all models — error on BERT-base is 26%, RoBERTa-base is 47%, MobileBERT is 30%, and DistilBERT is 36%. While there are some settings where software measurements is accurate (for example, the error is only 2.7% for RoBERTa-base model with batch size 2), it is not accurate in general.

Figure 2b shows that the standard deviations for software energy measurements are twice as large as that of hardware-based energy measurements. Large deviations for different runs of a model in the same setting makes the measurements unreliable.

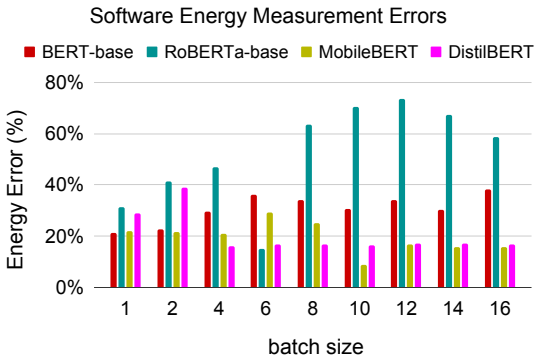
The main takeaway here that existing software-based energy measurements can be substantially inaccurate. However, they are more convenient to estimate energy consumption compared to using hardware power meters. Going forward, we need to design more accurate software measurements that come close to the ground truth.

3.2 Energy Measurements Using Hardware Agnostic Parameters Is Suboptimal

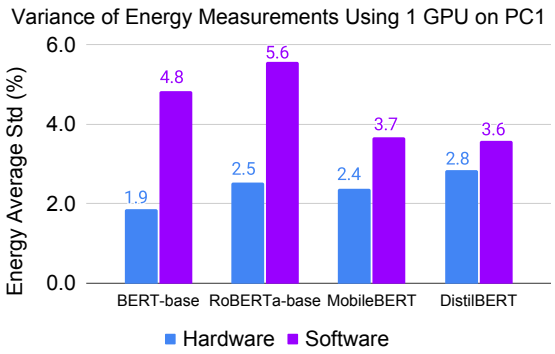
Why are existing software-based energy measurements (Strubell et al., 2019; Henderson et al., 2020) not accurate? The software-based energy model computes energy by aggregating resource usage as follows: $e_{total} = PUE \sum_p (p_{dram} e_{dram} + p_{cpu} e_{cpu} + p_{gpu} e_{gpu})$, where $p_{resource}$ [3] are the percentages of each system resource used by the attributable processes relative to the total in-use resources and $e_{resource}$ is the energy usage of that resource. The constant for power usage effectiveness (PUE) compensates for extra energy used to cool or heat data centers.

There are two potential problems in this linear energy model. First, different hardware devices (e.g. different CPU or GPU models, different number of GPUs connected, etc.) can have different cooling or heating effects causing large variations in the amounts of energy consumed. However, the energy model uses the PUE constant as a hardware agnostic parameter, which does not account for such differences in device specifications. This makes the final energy measurements less reliable. Second, assigning energy credits based on process resources is not always reliable. CPUs and GPUs often have power lags, power distortions, and tail energy especially during starting new processes or finishing existing processes (Burtscher et al., 2014;

[3] $resources$ can be $dram$, cpu, gpu



(a) Errors when using software-based energy measurements for the 4 studied models. We use the hardware power meter as the ground truth energy, and compute the error as the energy differences percentage of the ground truths.



(b) Standard deviations of both energy measured using hardware power meter and using software energy estimates. We compute the standard deviation across 10 runs for the 4 studied models.

Figure 2: Accuracy and robustness comparison between hardware and software based energy measurements. We use 1 GPU on PC1 for all the experiments.

Krzywda et al., 2018).

We conducted two empirical experiments to study these problems: (1) measuring the energy consumption of running the 4 NLP models on two different machines – **PC1** and **PC2**. The detailed device information is described in §2. (2) Use two GPUs on **PC1** to perform inference for the 4 NLP models instead of a single GPU.

Figure 3 shows that the energy errors are prominent when using two GPUs for inference compare to one-GPU setting or using a different GPU model. This is likely because the linear energy estimate model cannot easily take into account the above energy factors (power lag, distortion and tail energy) that affect GPU resources usage. Variable PUE can possibly address this, but that requires careful calibration based on the ground truth energy from the hardware power meters. Figure 4 shows the standard deviation when using existing software-based energy measurements.

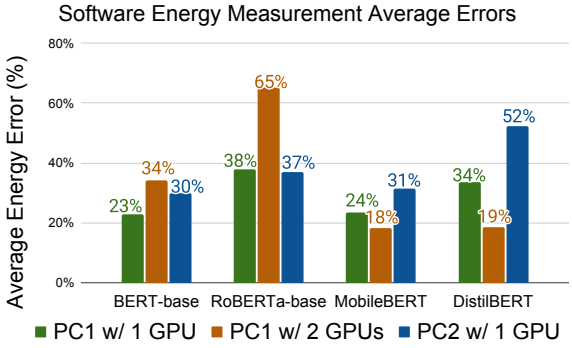
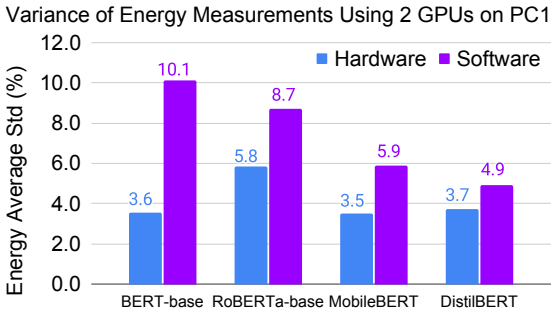


Figure 3: Average energy error of the software measurements for the 4 studied models using different hardware device configurations. The error patterns are different across all 3 settings, for example, (1) using two GPUs (instead of one) on the same machine can cause more errors; (2) using the same number of GPUs but with different hardware specifications may lead to different energy errors. (i.e., compare using 1 GPU on PC2 to 1 GPU on PC1)

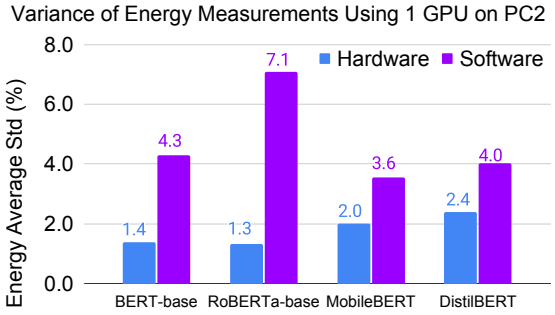
3.3 Software-Based Energy Measurements Can Lead to Bad Design Choices

The inaccuracy and robustness issues in software-based measurements can adversely impact model choices when considering energy and effectiveness trade-offs. To demonstrate this we consider two decision problems. One where we want to choose between BERT-base with RoBERTa-base, and another problem where we want to choose between MobileBERT and DistilBERT. Table 2 summarizes the performance scores of these models on the SQuAD 1.1 QA dataset. Figure 5a shows that RoBERTa-base correctly answers an additional 2.2% questions over BERT-base but it incurs 13% more energy on average. Similarly, MobileBERT answers 3.5% more questions correctly with 13% more energy budget compared to DistilBERT. Moreover, for MobileBERT and DistilBERT, batching questions help close the relative gap of energy costs.

If we instead use software-based energy measurements, however, presents a misleading picture. According to software energy measurements shown in Figure 5b, RoBERTa even consumes less energy than BERT (batch sizes 6 and 8), and MobileBERT

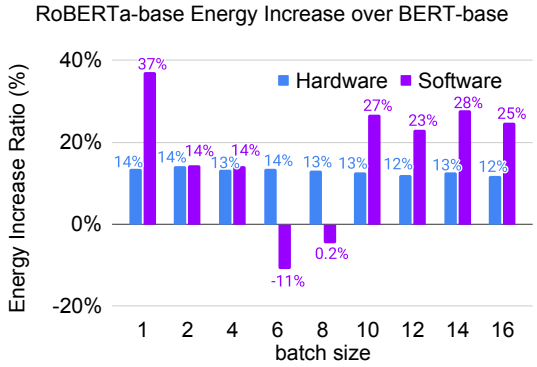


(a) Standard deviations of the hardware power meter measurements and software energy measurements using 2 GPUs on PC1 to perform inference.

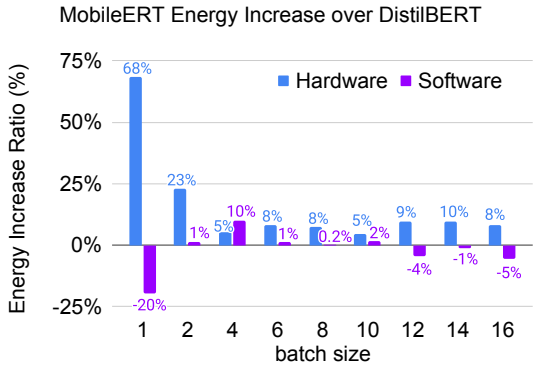


(b) Standard deviations of the hardware power meter measurement and software energy measurements 1 GPU on PC2 to perform inference.

Figure 4: Comparing the standard deviation in energy estimates under different hardware device configurations.



(a) Energy increase ratios from DistilBERT to MobileBERT.



(b) Energy increase ratios from BERT-base to RoBERTa-base.

Figure 5: Energy increase ratio comparison using hardware and software measurements.

can be more energy efficient than DistillBERT for many batch sizes (1, 12, 14, 16). Neither conclusion is true.

Model	EM	F1-score
BERT-base	80.8	88.2
RoBERTa-base	83.0	90.4
MobileBERT	82.6	90.0
DistilBERT	79.1	86.8

Table 2: SQuAD 1.1 task performance scores of the 4 studied models.

3.4 Interactions between Inference Latency and Energy Consumption Are Non-trivial

With the more accurate hardware energy measurements, we investigate the relationship between latency and energy consumption. In particular, we correlate the model energy consumption with its inference latency and task-specific performance. Note that, in this section, to better characterize the model inference latency and energy interactions, we do not use the end to end latency and energy numbers. Instead, we focus on the model prediction process, i.e. right before the model runs prediction and after the model finishes the prediction of all examples.

Figure 6 shows the inference latency speedup versus energy savings of MobileBERT and DistilBERT models over the RoBERTa-base model. We can see that smaller batch sizes (< 10) give more energy benefits compared to latency improvement, but as the inference batch size increases, the latency and energy savings are approximately proportional. This is beneficial to mobile settings where smaller batch sizes happen more frequently (e.g., users ask a question at a time instead of asking many questions simultaneously).

4 Related Work and Discussion

Energy estimation is an important research topic in both the machine learning and system community.

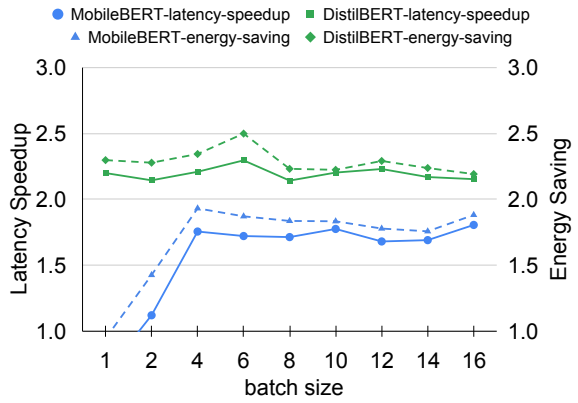


Figure 6: Latency speedup versus energy savings. All numbers are relative to the RoBERTa-base model. We report the hardware based energy values. Since the *experiment-impact-tracker* software does not sample sufficient energy values, we cannot extract the software energy for the prediction process only, hence omit comparison.

We discuss two threads of research related to energy estimation for NLP models:

Energy estimation in machine learning and NLP. Henderson et al. (2020) use a software framework called *experiment-impact-tracker* to report the aggregated energy of benchmark programs. The *experiment-impact-tracker* collects hardware resources statistics, and uses a simple linear model (Strubell et al., 2019) to estimate the total energy where the coefficients are fixed to a constant without considering the actual hardware device configurations. We have shown in the experiments that such software based energy estimation methods are neither accurate nor robust. We recommend using hardware power meters to measure the energy consumption, and then possibly calibrate the software energy values. (Zhou et al., 2020) presents an energy efficient benchmark for NLP models. However, they only report the time (hours) and cost (dollars) for training and testing NLP models, the actual energy numbers remain unknown. The Green AI (Schwartz et al., 2019) work suggests using metrics like floating point operations (FPO) to measure energy efficiency. However, Henderson et al. (2020) argues such metrics alone cannot accurately reflect energy consumption. García-Martín et al. (2019) provide a comprehensive survey of energy estimation methods in machine learning, but no energy measurements for NLP models were reported.

Energy modeling for systems and applications. Energy estimation for battery powered devices such as mobile phones is critical since mobile applications utility can be limited by the battery life. Previous work (Pathak et al., 2011, 2012; Yoon et al., 2012; Cao et al., 2017) study various fine-grained system-level power modeling and profiling techniques to help understand energy drain of applications. NLP models essentially power many emerging applications such as personal assistants with mobile intelligence. However, the energy implications for these NLP models are not studied. It is unclear how to apply the existing energy estimation methods for mobile applications to NLP models. We believe it is important to understand the computational semantics in NLP models before leveraging these existing power modeling methods. Using fine-grained power estimation models and profiling techniques could further improve our understanding of how NLP models consume energy and is an interesting future work.

Limitations. In this work, we collect the energy values every 1 second, which can reflect the total amount of energy consumed for batched inferences that often last over 10 seconds. However, if one needs to understand the energy spent inside the model for a single inference, the energy values are still coarse-grained, we will explore fine-grained energy measurement solutions to study this issue in the near future. More complex energy issues like tail power, energy distortion (Burtscher et al., 2014; Pathak et al., 2011) also affect hardware power meters if analyzing the energy spent inside the model. Further, it is not clear yet which machine component (CPU, GPU or memory reads/writes) takes how much energy for the NLP models. We leave this to future work.

5 Conclusions

As NLP models keep getting larger, reducing the energy impact of deploying these models is critical. Recent work has enabled estimating and tracking the energy of these NLP models. These works design a software-based technique to estimate energy consumption by tracking resource utilization. However, we show that currently used software-based measurements method is not accurate. We use a hardware power meter to accurately measure energy and find that this measurement method has an average error of 20% and can lead to making inaccurate design choices. Going forward, we hope this paper encourages the NLP community to build on current systems research to design more accurate

energy models that take into account the underlying power dynamics and device variabilities.

References

Global Warming of 1.5 °C.

WattsUp Meter Pro.

Martin Burtscher, Ivan Zecena, and Ziliang Zong. 2014. Measuring GPU Power with the K20 Built-in Sensor. In *Proceedings of Workshop on General Purpose Processing Using GPUs*, GPGPU-7, pages 28–36, New York, NY, USA. Association for Computing Machinery.

Yi Cao, Javad Nejati, Muhammad Wajahat, Aruna Balasubramanian, and Anshul Gandhi. 2017. Deconstructing the Energy Consumption of the Mobile Page Load. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 1(1):6:1–6:25.

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.

Eva García-Martín, Crefeda Faviola Rodrigues, Graham Riley, and Håkan Grahn. 2019. Estimation of energy consumption in machine learning. *Journal of Parallel and Distributed Computing*, 134:75–88.

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. *arXiv:2002.05651 [cs]*.

Jakub Krzywda, Ahmed Ali-Eldin, Trevor E. Carlson, Per-Olov Östberg, and Erik Elmroth. 2018. Power-performance tradeoffs in data center servers: DVFS, CPU pinning, horizontal, and vertical scaling. *Future Generation Computer Systems*, 81:114–128.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. *arXiv:1907.11692 [cs]*.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d\textquotesingle Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8026–8037. Curran Associates, Inc.

Abhinav Pathak, Y. Charlie Hu, and Ming Zhang. 2012. Where is the energy spent inside my app? fine grained energy accounting on smartphones with Eprof. In *Proceedings of the 7th ACM european conference on Computer Systems*, EuroSys '12, pages 29–42, New York, NY, USA. Association for Computing Machinery.

Abhinav Pathak, Y. Charlie Hu, Ming Zhang, Paramvir Bahl, and Yi-Min Wang. 2011. Fine-grained power modeling for smartphones using system call tracing. In *Proceedings of the sixth conference on Computer systems*, EuroSys '11, pages 153–168, New York, NY, USA. Association for Computing Machinery.

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.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

David Rolnick, Priya L. Donti, Lynn H. Kaack, Kelly Kochanski, Alexandre Lacoste, Kris Sankaran, Andrew Slavin Ross, Nikola Milojevic-Dupont, Natasha Jaques, Anna Waldman-Brown, Alexandra Luccioni, Tegan Maharaj, Evan D. Sherwin, S. Karthik Mukkavilli, Konrad P. Kording, Carla Gomes, Andrew Y. Ng, Demis Hassabis, John C. Platt, Felix Creutzig, Jennifer Chayes, and Yoshua Bengio. 2019. Tackling Climate Change with Machine Learning. *arXiv:1906.05433 [cs, stat]*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. *arXiv:1910.01108 [cs]*.

Roy Schwartz, Jesse Dodge, Noah A. Smith, and Oren Etzioni. 2019. Green AI. *arXiv:1907.10597 [cs, stat]*.

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

Zhiqing Sun, Hongkun Yu, Xiaodan Song, Renjie Liu, Yiming Yang, and Denny Zhou. 2020. MobileBERT: a Compact Task-Agnostic BERT for Resource-Limited Devices. *arXiv:2004.02984 [cs]*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. HuggingFace's Transformers: State-of-the-art Natural Language Processing. *arXiv:1910.03771 [cs]*.

Chanmin Yoon, Dongwon Kim, Wonwoo Jung, Chulkoo Kang, and Hojung Cha. 2012. AppScope: application energy metering framework for android smartphones using kernel activity monitoring. In *Proceedings of the 2012 USENIX conference on Annual Technical Conference*, USENIX ATC'12, page 36, USA. USENIX Association.

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, and William Yang Wang. 2020. HULK: An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing. *arXiv:2002.05829 [cs]*.