

# HULK: An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing

Xiyou Zhou, Zhiyu Chen, Xiaoyong Jin, William Yang Wang

Department of Computer Science, University of California Santa Barbara

{xiyou, zhiyuchen, x-jin, william}@cs.ucsb.edu

## Abstract

Computation-intensive pretrained models have been taking the lead of many natural language processing benchmarks such as GLUE (Wang et al., 2018). However, energy efficiency in the process of model training and inference becomes a critical bottleneck. We introduce HULK, a multi-task energy efficiency benchmarking platform for responsible natural language processing. With HULK, we compare pretrained models’ energy efficiency from the perspectives of time and cost. Baseline benchmarking results are provided for further analysis. The fine-tuning efficiency of different pretrained models can differ a lot among different tasks and fewer parameter number does not necessarily imply better efficiency. We analyzed such phenomenon and demonstrate the method of comparing the multi-task efficiency of pretrained models. Our platform is available at <https://sites.engineering.ucsb.edu/~xiyou/hulk/>.

## 1 Introduction

Environmental concerns of machine learning research has been rising as the carbon emission of certain tasks like neural architecture search reached an exceptional “ocean boiling” level (Strubell et al., 2019). Increased carbon emission has been one of the key factors to aggravate global warming<sup>1</sup>. Research and development process like parameter search further increase the environment impact. When using cloud-based machines, the environment impact is strongly correlated with budget.

The recent emergence of leaderboards such as SQuAD (Rajpurkar et al., 2016), GLUE (Wang et al., 2018) and SuperGLUE (Wang et al., 2019) has greatly boosted the development of advanced models in the NLP community. Pretrained models

have proven to be the key ingredient for achieving state of the art in conventional metrics. However, such models can be extremely expensive to train. For example, XLNet-Large (Yang et al., 2019) was trained on 512 TPU v3 chips for 500K steps, which costs around 61,440 dollars<sup>2</sup>, let alone staggeringly large carbon emission.

Moreover, despite impressive performance gain, the fine-tuning and inference efficiency of NLP models remain under-explored. As recently mentioned in a tweet<sup>3</sup>, the popular AI text adventure game *AI Dungeon* has reached 100 million inferences. The energy efficiency of inference cost could be critical to both business planning and environment impact.

Previous work (Schwartz et al., 2019; Dodge et al., 2019) on this topic proposed new metrics like FPO (floating point operations) and new practice to report experimental results based on computing budget. Other benchmarks like (Coleman et al., 2017) and (Mattson et al., 2019) compares the efficiency of models on the classic reading comprehension task SQuAD and machine translation tasks. However, there has not been a concrete or practical reference for accurate estimation on NLP model pretraining, fine-tuning and inference considering multi-task energy efficiency.

Energy efficiency can be reflected in many metrics including carbon emission, electricity usage, time consumption, number of parameters and FPO as shown in (Schwartz et al., 2019). Carbon emission and electricity are intuitive measures yet either hard to track or hardware-dependent. Number of parameters does not reflect the actual cost for model training and inference. FPO is steady for models but cannot be directly used for cost estimation. Here in order to provide a practical reference

<sup>1</sup>Source: <https://climate.nasa.gov/causes/>

<sup>2</sup>Source: <https://bit.ly/301qUMo>

<sup>3</sup>Source: <https://bit.ly/2GAFBNO>

Model	Hardware	Time	Cost	Params
BERT <sub>BASE</sub> (Devlin et al., 2018)	4 TPU Pods	4 days	\$1,728	108M
BERT <sub>LARGE</sub> (Devlin et al., 2018)	16 TPU Pods	4 days	\$6,912	334M
XLNet <sub>BASE</sub> (Yang et al., 2019)	—	—	—	117M
XLNet <sub>LARGE</sub> (Yang et al., 2019)	512 TPU v3	2.5 days	\$61,440	361M
RoBERTa <sub>BASE</sub> (Liu et al., 2019)	1024 V100 GPUs	1 day	\$75,203	125M
RoBERTa <sub>LARGE</sub> (Liu et al., 2019)	1024 V100 GPUs	1 day	\$75,203	356M
ALBERT <sub>BASE</sub> (Lan et al., 2019)	64 TPU v3	—	—	12M
ALBERT <sub>LARGE</sub> (Lan et al., 2019)	—	—	—	18M
ALBERT <sub>XLARGE</sub> (Lan et al., 2019)	—	—	—	59M
ALBERT <sub>XXLARGE</sub> (Lan et al., 2019)	1024 TPU v3	32 hours	\$65,536	223M
DistilBERT* (Sanh et al., 2019)	8×16G V100 GPU	90 hours	\$2203.2	66M

Table 1: Pretraining costs of baseline models. Hardware and pretraining time are collected from original papers, with which costs are estimated with current TPU price at \$8 per hour with 4 core TPU v3 chips and V100 GPU at \$3.06 per hour. DistilBERT model is trained upon a pretrained BERT model. Parameter numbers are estimated using the pretrained models implemented in the Transformers (<https://github.com/huggingface/transformers>) library (Wolf et al., 2019), shown in million.

for model selection for real applications, especially model development outside of academia, we keep track of the time consumption and actual budget for comparison. Cloud based machines are employed for cost estimation as they are easily accessible and consistent in hardware configuration and performance. In the following sections, we would use time and cost to denote the time elapsed and the actual budget in model pretraining / training / inference.

In most NLP pretrained model setting, there are three phases: pretraining, fine-tuning and inference. If a model is trained from scratch, we consider such model has no pretraining phase but fine-tuned from scratch. Typically pretraining takes several days and hundreds of dollars, according to Table 1. Fine-tuning takes a few minutes to hours, costing a lot less than pretraining phase. Inference takes several milli-seconds to seconds, costing much less than fine-tuning phase. Meanwhile, pretraining is done before fine-tuning once for all, while fine-tuning could be performed multiple times as training data updates. Inference is expected to be called numerous times for downstream applications. Such characteristics make it an intuitive choice to separate different phases during benchmarking.

Our HULK benchmark, as shown in Figure 1, utilizes several classic datasets that have been widely adopted in the community as benchmarking tasks to benchmark energy efficiency and compares pretrained models in a multi-task fashion. The tasks include natural language inference task

MNLI (Williams et al., 2017), sentiment analysis task SST-2 (Socher et al., 2013) and Named Entity Recognition Task CoNLL-2003 (Sang and De Meulder, 2003). Such tasks are selected to provide a thorough comparison of end-to-end energy efficiency in pretraining, fine-tuning and inference.

With the HULK benchmark, we quantify the energy efficiency of model pretraining, fine-tuning and inference phase by comparing the time and cost they require to reach certain overall task-specific performance level on selected datasets. The design principle and benchmarking process are detailed in section 2. We also explore the relation between model parameter and fine-tuning efficiency and demonstrate consistency of energy efficiency between tasks for different pretrained models.

## 2 Benchmark Overview

For pretraining phase, the benchmark is designed to favor energy efficient models in terms of time and cost that each model takes to reach certain multi-task performance pretrained from scratch. For example, we keep track of the time and cost of a BERT model pretrained from scratch. After every thousand of pretraining steps, we clone the model for fine-tuning and see if the final performance can reach our cut-off level. When the level is reached, time and cost for pretraining is used for comparison. Models faster or cheaper to pretrain are recommended.

For fine-tuning phase, we consider the time and cost each model requires to reach certain multi-

	CoNLL 2003	MNLI	SST-2
Train Size	14,041	392,702	67,349
Dev Size	3,250	19,647	872
Cut-off	91	85	90
Metric	F1	Acc	Acc
SOTA	93.5	91.85	97.4

Table 2: Dataset Information

task performance fine-tuned from given pretrained models because for each single task with different difficulty and instance number, the fine-tuning characteristics may differ a lot. When pretrained models are used to deal with non-standard downstream task, especially ad hoc application in industry, the training set’s difficulty cannot be accurately estimated. Therefore, it’s important to compare the multi-task efficiency for model choice.

For inference phase, the time and cost of each model making inference for single instance on multiple tasks are considered in the similar fashion as the fine-tuning phase.

## 2.1 Dataset Overview

The datasets we used are widely adopted in NLP community. Quantitative details of datasets can be found in Table 2. The selected tasks are shown below:

**CoNLL 2003** The Conference on Computational Natural Language Learning (CoNLL-2003) shared task concerns language-independent named entity recognition (Sang and De Meulder, 2003). The task concentrates on four types of named entities: persons, locations, organizations and other miscellaneous entities. Here we only use the English dataset. The English data is a collection of news wire articles from the Reuters Corpus. Result is reflected as F1 score considering the label accuracy and recall on dev set.

**MNLI** The Multi-Genre Natural Language Inference Corpus (Williams et al., 2017) is a crowdsourced collection of sentence pairs with textual entailment annotations. Given a premise sentence and a hypothesis sentence, the task is to predict whether the premise entails the hypothesis (entailment), contradicts the hypothesis (contradiction), or neither (neutral). The premise sentences are gathered from ten differ-

ent sources, including transcribed speech, fiction, and government reports. The accuracy score is reported as the average of performance on matched and mismatched dev sets.

**SST-2** The Stanford Sentiment Treebank (Socher et al., 2013) consists of sentences from movie reviews and human annotations of their sentiment. The task is to predict the sentiment of a given sentence. Following the setting of GLUE, we also use the two-way (positive/negative) class split, and use only sentence-level labels.

The tasks are selected based on how representative the dataset is. CoNLL 2003 has been a widely used dataset for named entity recognition and actually requires output of token level labeling. NER is a core NLP task and CoNLL 2003 has been a classic dataset in this area. SST-2 and MNLI are part of the GLUE benchmark, representing sentence level labeling tasks. SST-2 has been frequently used in sentiment analysis across different generations of models. MNLI is a newly introduced large dataset for natural language inference. The training time for MNLI is relatively long and the task requires a lot more training instances. We select the three tasks for a diverse yet practical benchmark for pretrained models without constrain the models to sentence level classification tasks. In addition, their efficiency differ significantly in the fine-tuning and inference phase. Such difference can still be reflected on the final score after normalization as shown in Table 3. Provided with more computing resource, we can bring in more datasets for even more thorough benchmarking in the future. We illustrate the evaluation criteria in the following subsection.

## 2.2 Evaluation Criteria

In machine learning model training and inference, slight parameter change can have subtle impact on the final result. In order to make a practical reference for pretrained model selection, we compare models’ end-to-end performance with respect to the pretraining time, pretraining cost, training time, training cost, inference time, inference latency and cost following the setting of (Coleman et al., 2017).

For pretraining phase, we design the process to explore how much computing resource is required to reach certain multi-task performance by fine-tuning after the pretraining. Therefore, during

## HULK

Save the world, one flop at a time.

An Energy Efficiency Benchmark Platform for Responsible Natural Language Processing

### Named Entity Recognition - CONLL 2003

Rank	Time to 90 Test F1	Model	Hardware	Framework
1 Nov 2019	90.26	BERT-Large-Cased BERT Baseline	GTX 2080Ti	Pytorch 0.3.1 post2
2 Nov 2019	155.43	RoBERTa-LARGE RoBERTa Baseline	GTX 2080Ti	Pytorch 0.3.1 post2

Figure 1: Screenshot of the leaderboard of website.

Datasets	CoNLL 2003		SST-2		MNLI		
Model	Time	Score	Time	Score	Time	Score	Overall Score
BERT <sub>BASE</sub>	43.43	2.08	207.15	0.45	N/A	0.00	2.53
BERT <sub>LARGE</sub>	90.26	1.00	92.45	1.00	9,106.72	1.00	3.00
XLNet <sub>BASE</sub>	67.14	1.34	102.45	0.90	7,704.71	1.18	3.42
XLNet <sub>LARGE</sub>	243.00	0.37	367.11	0.25	939.62	9.69	10.31
RoBERTa <sub>BASE</sub>	70.57	1.28	38.45	2.40	274.87	7.14	10.82
RoBERTa <sub>LARGE</sub>	155.43	0.58	57.65	1.60	397.12	22.93	25.11
ALBERT <sub>BASE</sub>	340.64	0.26	2,767.90	0.03	N/A	0.00	0.29
ALBERT <sub>LARGE</sub>	844.85	0.11	3,708.49	0.02	N/A	0.00	0.13

Table 3: Multi-task Baseline Fine-tuning Costs. Time is given in seconds and score is computed by the division of  $\text{Time}_{\text{BERT}_{\text{LARGE}}}/\text{Time}_{\text{model}}$ . The experiments are conducted on a single GTX 2080 Ti GPU following the evaluation criteria. The overall score is computed by summing up scores of each individual task. For cost based leaderboards, we also use the budget to compute a new score for each task and summarize similarly. “N/A” means fail to reach the given performance after 5 epochs.

model pretraining, after a number of steps, we use the half-pretrained model for fine-tuning and see if the fine-tuned model can reach our cut-off performance. When it does, we count the time and cost in the pretraining process for benchmarking and analysis.

For fine-tuning phase, we want to compare the general efficiency of pretrained model reaching cut-off performance on selected dataset. During fine-tuning, we evaluate the half-fine-tuned model on development set after a certain number of steps. When the performance reach our cut-off performance, we count the time and cost in this fine-tuning process for benchmarking and analysis. To be specific, for a single pretrained model, the efficiency score on different tasks is defined as the sum of normalized time and cost. Here we normalize the time and cost because they vary dramatically between tasks. In order to simplify the process, we

compute the ratio of BERT<sub>LARGE</sub>’s time and cost to that of each model as the normalized measure as shown in Table 3 and Table 4.

For inference phase, we follow the principles in fine-tuning except we use the time and cost of inference for benchmarking.

### 2.3 Performance Cut-off Selection

The selection of performance cutoff could be very critical because we consider certain models being qualified after reaching certain performance on development set. Meanwhile, certain tasks can reach a “sweet point” where after relatively smaller amount of training time, the model reaches performance close to the final results despite negligible difference. We select the cut-off performance threshold by observing the recent state-of-the-art performance on selected tasks.

Datasets	CoNLL 2003		SST-2		MNLI		
Model	Time	Score	Time	Score	Time	Score	Overall Score
BERT <sub>BASE</sub>	2.68	3.18	2.70	3.13	2.67	3.19	9.5
BERT <sub>LARGE</sub>	8.51	1.00	8.46	1.00	8.53	1.00	3.00
XLNet <sub>BASE</sub>	5.16	1.65	5.01	1.69	5.10	1.67	5.01
XLNet <sub>LARGE</sub>	14.84	0.57	14.69	0.58	15.27	0.56	1.71
RoBERTa <sub>BASE</sub>	2.65	3.21	2.68	3.16	2.70	3.16	9.53
RoBERTa <sub>LARGE</sub>	8.35	1.02	8.36	1.01	8.70	0.98	3.01
ALBERT <sub>BASE</sub>	2.65	3.21	2.68	3.18	2.72	3.14	9.53
ALBERT <sub>LARGE</sub>	8.49	1.00	8.44	1.00	8.78	0.97	2.97

Table 4: Multi-task Baseline Inference Costs. Time is given in milliseconds and score is computed by the division of  $\text{Time}_{\text{BERT}_{\text{LARGE}}} / \text{Time}_{\text{model}}$ . The experiments are conducted on a single GTX 2080 Ti GPU following the evaluation criteria similar to fine-tuning part. It’s clear that the inference time between tasks is more consistent compared to fine-tuning phase.

## 2.4 Submission to Benchmark

Submissions can be made to our benchmark through sending code and results to our HULK benchmark CodaLab competition<sup>4</sup> following the guidelines in both our FAQ part of website and competition introduction. We require the submissions to include detailed end-to-end model training information including model run time, cost (cloud based machine only), parameter number and part of the development set output for result validation. A training / fine-tuning log including time consumption and dev set performance after certain steps is also required. For inference, development set output, time consumption and hardware / software details should be provided. In order for model reproducibility, source code is required.

## 3 Baseline Settings and Analysis

For computation-heavy tasks, we adopt the reported resource requirements in the original papers as the pretraining phase baselines.

For fine-tuning and inference phase, we conduct extensive experiments on given hardware (GTX 2080Ti GPU) with different model settings as shown in Table 3 and Table 4. We also collect the development set performance with time in fine-tuning to investigate in how the model are fine-tuned for different tasks.

In our fine-tuning setting, we are given a specific hardware and software configuration, we adjust the hyper-parameter to minimize the time required for fine-tuning towards cut-off performance. For example, we choose proper batchsize and learning rate

for BERT<sub>BASE</sub> to make sure the model converges and can reach expected performance as soon as possible with parameter searching.

As shown in Figure 2, the fine-tuning performance curve differs a lot among pretrained models. The x-axis denoting time consumed is shown in log-scale for better comparison of different models. None of the models actually take the lead in all tasks. However, if two pretrained models are in the same family, such as BERT<sub>BASE</sub> and BERT<sub>LARGE</sub>, the model with smaller number of parameters tend to converge a bit faster than the other in the NER and SST-2 task. In the MNLI task, such trend does not apply possibly due to increased difficulty level and training instance number which favor larger model capacity.

Even though ALBERT model has a lot less parameters than BERT, according to Table 1, the fine-tuning time of ALBERT model is significantly more than BERT models. This is probably because ALBERT uses large hidden size and more expensive matrix computation. The parameter sharing technique actually makes it harder to fine-tune the model. RoBERTa<sub>LARGE</sub> model relatively stable in all tasks.

## 4 Related Work

GLUE benchmark (Wang et al., 2018) is a popular multi-task benchmarking and diagnosis platform providing score evaluating multi-task NLP models considering multiple single task performance. SuperGLUE (Wang et al., 2019) further develops the task and enriches the dataset used in evaluation, making the task more challenging. These

<sup>4</sup>The CodaLab competition is accessible from the website.



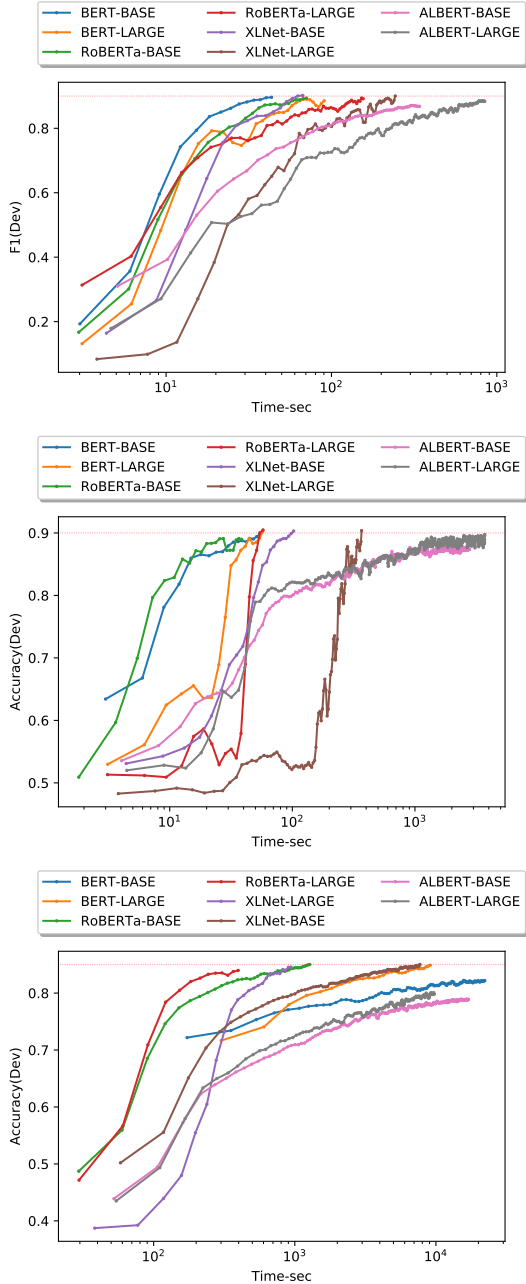


Figure 2: The comparison between different pretrained models for CoNLL 2003, SST-2 and MNLI datasets trained on a single GTX 2080Ti GPU. The curves are smoothed by computing average with 2 adjacent data points. The experiments are conducted by selecting hyper-parameters to minimize the time consumption yet making sure the model can converge after certain amount of time. Results are demonstrated using performance on development score after certain steps fine-tuned on the training dataset.

multi-task benchmarks does not take computation efficiency into consideration but still innovates the development of pretrained models.

MLPerf (Mattson et al., 2019) compares training and inference efficiency from hardware perspective, providing helpful resources on hardware selection and model training. Their benchmark is limited to focusing on several typical applications including image classification and machine translation.

Previous work (Schwartz et al., 2019; Dodge et al., 2019) on related topic working towards “Green AI” proposes new metrics like FPO and new principle in efficiency evaluation. We further make more detailed and practical contributions towards model energy efficiency benchmarking.

Other work like DAWN Benchmark (Coleman et al., 2017) looks into the area of end-to-end model efficiency comparison for both computer vision and NLP task SQuAD. The benchmark does not compare multi-task efficiency performance and covered only one NLP task.

The *Efficient NMT* shared task of The 2nd Workshop on Neural Machine Translation and Generation proposed efficiency track to compare neural machine translation models’ inference time. Our platform covers more phases and support multi-task comparison.

## 5 Conclusion

We developed the HULK platform focusing on the energy efficiency evaluation of NLP models based on their end-to-end performance on selected NLP tasks. The HULK platform compares models in pretraining, fine-tuning and inference phase, making it clear to follow and propose more training and inference efficient models. We have compared the fine-tuning efficiency of given models during baseline testing and demonstrated more parameters lead to slower fine-tuning when using same model but does not hold when model changes. We expect more submissions in the future to flourish and enrich our benchmark.

## Acknowledgments

This work is supported by the Institute of Energy Efficiency (IEE) at UCSB’s seed grant in Summer 2019 to improve the energy efficiency of AI and machine learning.<sup>5</sup>

<sup>5</sup><https://iee.ucsb.edu/news/making-ai-more-energy-efficient>

## References

- Cody Coleman, Deepak Narayanan, Daniel Kang, Tian Zhao, Jian Zhang, Luigi Nardi, Peter Bailis, Kunle Olukotun, Chris Ré, and Matei Zaharia. 2017. Dawnbench: An end-to-end deep learning benchmark and competition. *Training*, 100(101):102.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
- Jesse Dodge, Suchin Gururangan, Dallas Card, Roy Schwartz, and Noah A Smith. 2019. Show your work: Improved reporting of experimental results. *arXiv preprint arXiv:1909.03004*.
- 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. *arXiv preprint arXiv:1909.11942*.
- 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 preprint arXiv:1907.11692*.
- Peter Mattson, Christine Cheng, Cody Coleman, Greg Diamos, Paulius Micikevicius, David Patterson, Hanlin Tang, Gu-Yeon Wei, Peter Bailis, Victor Bitorf, et al. 2019. Mlperf training benchmark. *arXiv preprint arXiv:1910.01500*.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*.
- Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*.
- Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.
- Roy Schwartz, Jesse Dodge, Noah A Smith, and Oren Etzioni. 2019. Green ai. *arXiv preprint arXiv:1907.10597*.
- Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.
- Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*.
- Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2019. Super-glue: A stickier benchmark for general-purpose language understanding systems. *arXiv preprint arXiv:1905.00537*.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.
- Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R’emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface’s transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.