Measure LLM knowledge via RIG (Raw Information Gain) over preliminarily crafted probes

Anonymous EMNLP submission

Abstract

We propose a novel high-level approach to analyze models in a different way: we could estimate amount of information receipted by a model using a crafted set of control statements. We introduce a new metrics RIG (raw information gain) in order to do so. Any LLM (large language model) could be considered a "black box" of compressed information. It is hard to measure what amount of information is stored inside the model regarding any domain. The contrast between the size of a trained model of around 43GB [1] compared to 15 trillion tokens of training data is staggering [2]. The other issue is to figure out where do the limits come from: is it an architectural constraint or is the limitation coming from the data used in training. So far the most common way to identify if the model is properly trained and contains necessary information is to put it through a list of benchmarks and the decision is based on either it's ranking or some educated guess of a score threshold. Keeping in mind that the most of those benchmarks become part of training data for upcoming models we face a vicious cycle of never ending benchmark creation. Taken into account constant size growth of both language models and datasets we face an challenge of losing a track of what is efficient and what is not to train models as well as simple scale of the datasets makes them almost impossible to supervise at all, what is an immense obstacle when we need to update any language model according to different environments those are implemented at and we need to bring ethical issues, actuality of the human knowledge and controversial statements altogether.

1 Introduction

1.1 Related Work

The starting point of this research was Shannon's (1948) "A Mathematical Theory of Communication". The key elements borrowed from his theory are the definition of a bit of information, the general communication system principles, and Shannon's definition of entropy. The training process of a language model fits the schema of general communication: the training dataset acts as an information source, a batch of token sequences as a message and the current state of the model acts as a receiver. We consider an LLM to be a receiving agent receiving a message and based on that its state is updated. Statistical language model theory is based on these features as well ((Croft and Lafferty, 2003)), yet there is one very important and unanswered question: "How much information does the text message contain?"

So far text data was treated according to Character encoding principles [3] as a sequence of characters which should be processed commonly through tokenization ((Manning et al., 2014)), word embedding ((Mikolov et al., 2013)) and then fed to a language model. If we measure how many bits a text message takes from a character encoding perspective, it does not represent the amount of information contained in this message. Token representation is more size-efficient ((Brisaboa et al., 2010),(Delétang et al., 2023)) but still, the question of "How much knowledge is contained in a token sequence?" is left unanswered.

The text itself is a form of natural language, used and created primarily by humans to exchange information with each other, however, every human perceives information from the text differently and this should not be discarded. There have been multiple studies based on human nature over the years about what is text for a person, how it is perceived, and what are efficient and non-efficient ways to use it. If we want to measure anyhow how much information a given piece of text contains, most aspects to be taken into account are unfortunately subjective such as the amount of information received

[1] https://huggingface.co/nitsuai/Meta-Llama-3-70B-Instruct-GGUF

[2] https://ai.meta.com/blog/meta-llama-3/

[3] https://en.wikipedia.org/wiki/Character_encoding

Model Size	Layers	Model Dim	Heads	Learning Rate
70 M	6	512	8	$1e-4$
160 M	12	768	12	$6e-4$
410 M	24	1024	16	$3e-4$
1.0 B	16	2048	8	$3e-4$
1.4 B	24	2048	16	$2e-4$
2.8 B	32	2560	32	$1.6e-4$

Table 1: Models from the Pythia suite and select hyperparameters.

by a person after hearing or reading a message is based on the person's prior knowledge, ability to understand the language and the vocabulary of the message and persons' mental and emotional condition. Each next word contained a different amount of information based on the prior context, and a recipient's knowledge, thus the expectancy of the following word could be different. Thereby we could not answer objectively if a given book was resourceful or not objectively ((Galanter, 1962)). The solution was to use some social mechanisms to evaluate the utility of a text example based on some majority votes or some expert evaluation.

1.2 Theoretical Approach

Whether people receive a message they approximate the upcoming word based on their knowledge and receive an amount of novel information if the upcoming word is unexpected, yet if the message loses coherence the word flow starts sounding like a random sequence with no utility value. If we could measure a perplexity amount for each word in a text sequence for a receiving person, we could estimate the informational value of the given text word by word.

CLM (Casual Language Modeling, (Radford et al., 2018)) task is very close to human natural perception of text: model gives us a probability distribution for an upcoming token based on the previous sequence (context) and state of the model (prior knowledge). That means that each following token in an input sequence makes some amount of general sense to the model, so the context changes the entropy for each further token distribution. Before any updates to model weights, the inference pass gives us contextual token distribution for each position in a sequence fed to the model. Disregarding the model architecture or size, the result would always be logits calculated through the inference pass.

We calculate Shannon entropy of the next token probability over vocabulary for each position with and without context. The difference between those entropy values for each token position would be the amount of information in bits brought by a token. We propose this metric as Raw Information Gain (RIG). It would be calculated like this:

$RIG_{token} = (-\sum_1^N p_k^* \cdot log_2(p_k^*)) - (-\sum_1^N p_k \cdot log_2(p_k))$;

where N is vocabulary size, p_k is a probability of the next token to be a k_{th} item over the vocabulary based on the context and p_k^* is a probability of the next token to be a k_{th} item without any context. If we sum up RIG_{token} for each token in the input sequence, we will get the amount of information that is perceived by the model. The probabilities could be calculated directly from the logits, but we need to perform two inference passes: one with a casual mask applied for the context-based probabilities and the second one with a diagonal mask restricting any context information except the token positioning. If we measure RIG for the same sequence for different states of the model or even different models we would receive different values and it would give us some high-level understanding of how much information the model "understands" from this sequence and that value is subjective to the sequence itself. For future reference, we set up some specific sequences which we call probes and evaluate RIG based on that.

2 Experimental setup

2.1 Hypothesis

Our guess is that by each communication stage (batch) the model updates its state via backward propagation, so the RIG value for the same probe should change over the training progress. Thus, while the model absorbs knowledge from the training dataset, a general understanding of a control text sequence (probe) should increase. If the original dataset has some corrupted data or new batches of data "flush" previously gained knowledge away RIG value should drop. Tracking this value over the training process might give us better steering than loss/validation metrics values, especially over the late stages of training, where loss looks like a plateau.

2.2 Model Suite

To make this analysis we are using a Pythia set of trained models ((Biderman et al., 2023)). Those models were trained on the same data in the same order and for the same amount of steps. All checkpoints are saved and available for download. We use the models of sizes 14M, 31M, 70M, 160M, 410M, 1M, 1.4B, 2.8B (parameters are listed at Table 1). Those models are trained on two versions of PiLE ((Gao et al., 2020)) dataset: base and deduplicated one. Since the deduplicated dataset is smaller but the models were trained on the same amount of steps, deduplicated versions of the model have seen some portions of the dataset more than once. We compare RIG value for these models on the same steps and refer to the metrics provided by the original Pythia paper to make some conclusions on optimal parameter choice and model training behavior.

2.3 General Domain Knowledge

We created a list of 50 general-domain short (average sequence length is about 50 tokens) probe pairs based on the TriviaQA dataset (Joshi et al., 2017) (examples are shown in Table 2). The first probe contains true knowledge and the second one contains false knowledge. We take a checkpoint make an inference pass for all the probes and average RIG value over probes for both truthful and false batches.
Implementation looks like that:

Algorithm 1: Evaluating RIG over model

for $i \in \{1, \ldots, N\}$ **do**
  model ← Download checkpoint(i);
  logits ← model(probes[]);
  apply diagonal mask to model;
  logits* ← model(probes[]);
  RIG ← shannon_entropy(logits*) -
    shannon_entropy(logits);
  track(i, RIG);
end

Despite the approach being straightforward it is worth mentioning, that PyTorch framework ((Imambi et al., 2021)) operates **-inf** value, which appears in logits at some point. It creates a conflict of **-inf** · **log(-inf)** which would result in a NaN value, since **log(-inf)** equals zero in PyTorch logic. **-inf** should be arbitrarily changed to any value before multiplying it by zero

2.4 Different Knowledge Complexity

In the second setup, we have two groups of 500 token probes regarding domain-specific knowledge (Physics, Chemistry, etc). The first group is based on 9th-grade educational material, while the second group consists of graduate-level sequences. Topics and described phenomenons are similar to each other (i.e. motion description either over basic kinematic equations or over differential equations). Our goal is to understand how much information is contained in the topics requiring larger background knowledge.

3 Results

3.1 False and True information difference

Intuition tells us that if we hear a message that contradicts our prior knowledge it means that the message contains either false information or new information. In both cases, the expectations of what we would receive would be different from the actual message, meaning that the conditional entropy of each following word based on the context would be higher. Up to some extent it is confirmed by our observations. If two statements are close in amount of tokens and one contains obvious truth and the other is obviously false we would have a noticeably higher RIG value for the false statement. However, if the statement is incorrect in a less obvious manner, the difference might not be notable (examples are given in Figure 1). Our guess is that the ability of incorrect statement detection (RIG spike) corresponds to the quality of the model since larger models were better at those comparisons of inconspicuous probe pairs. That phenomenon could be used to track a model's ability to interpret complex contradictions. The more sophisticated the false injection is, the higher the model quality should be for this RIG difference to occur. If we have a set of true-false probes of increasing complexity we can easily evaluate what model distinguishes. A pair of true-false probes might give us different absolute RIG values for different models, but the relative relation between those could always be estimated. Another important observation on those pairs is that probes should be more or less balanced in terms of token length, otherwise the difference in length might completely negate this RIG difference. Nevertheless, some false probes resulted in higher RIG values despite being noticeably shorter than their truthful counterpart. Our recommendation is to measure each probe pair on a high-quality model

Truthful entailment probe	False entailment probe
Flu could cause a serious toll on a person's health if left unattended	Flu doesn't need any special attention and goes away on its own
There are eight planets in our Solar System	There are nine planets in our Solar System
Why do veins appear blue? Veins appear blue because blue light does not penetrate deeply into human tissue	Why do veins appear blue? It's actually because deoxygenated blood is blue
What would happen if you were struck by a penny dropped from the top of the Empire State Building?You would feel a light impact if you were struck by a penny dropped from the Empire State Building	What would happen if you were struck by a penny dropped from the top of the Empire State Building? If you were struck by a penny dropped from the top of the Empire State Building
Has MSG been scientifically proven to be harmful to humans in small doses? No, there are no rigorous scientific studies showing that MSG is harmful to humans in small doses	Has MSG been scientifically proven to be harmful to humans in small doses? Yes. MSG has been proven to cause headaches and nausea
What are examples of films that remain banned by British censors? The film Mikey remains banned in Britain	What are examples of films that remain banned by British censors? The Evil Dead was banned until 1990
How long should you wait before filing a missing person report? You can make a missing person report as soon as you think a person is missing	How long should you wait before filing a missing person report? You should wait 48 hours before filing a missing person report

Table 2: Examples of TriviaQA derived probes containing a truthful and false answer

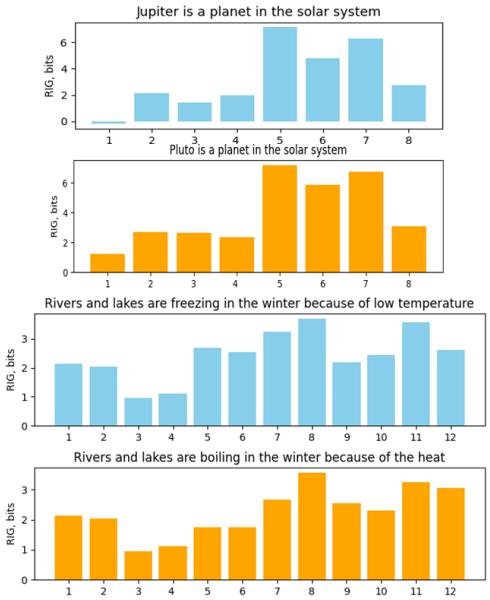

Figure 1: RIG per token distribution for 'obvious' and 'tricky' True and False probes. Total RIG values for 'Jupiter' and 'Pluto' probe are **26.3 bits** and **31.8 bits** correspondingly. Total RIG values for 'cold river' and 'hot river' probes are **29.2 bits** and **27.1 bits**. This RIG distributions are measured on 2.8b non-deduped Pythia model (checkpoint 141 of 141)

before you utilize them for training of any model.

3.2 Training curves

We noticed three distinctive features on this set of models. The first one is the different behavior of models on a large scale (Figure 2). 31M, 70M, and 70M-deduped models have easily detectable decline in RIG value even after 20k steps of training. That is causal evidence of models inability to assimilate new knowledge coming from the new batches of data. We assume that it's a very easy way to detect catastrophic forgetting ((Kirkpatrick et al., 2017)) phenomenon. Strictly speaking the curve in our experiment only shows RIG decline regarding the knowledge represented in probes, yet those probes were not domain-specific but related to the general knowledge domain. Benchmark scores for those models were also decreasing and that was represented in the original Pythia report ((Biderman et al., 2023))and that correlates with our observations. Some models (160M, 160M deduped, 410M, 1B) were reaching a plateau, meaning that general domain knowledge stopped increasing around 100k training steps. Some models showed continuous RIG growth (14M, 1.4B, 2.8B, 410M deduped, 1B deduped, 1.4B deduped) meaning those were clearly undertrained even at 141000 steps. All in

all, it is easy to observe metrics to understand in order to make a decision if the model should be kept training or not, which is easier to follow than recommendations from ((Hoffmann et al., 2022)).

Another observation is that the larger the model the flatter the RIG curve becomes. 1.4B and 2.8B models have notably smaller RIG growth over the first 20k steps. It's worth mentioning that plotted curves are represented using the full-fidelity bucketing method over the X-axis with a smoothing coefficient of 0.5. To negate fluctuations of RIG value over each step that is a necessary approximation The third observation is regarding the dataset. The unique feature of Pythia (Biderman et al., 2023) suite is that each step of all models was trained on the same data. That implies if we notice sufficient RIG decline happens for multiple models on a single step, the data quality used in that step was low. For example, step 90k was incredibly harmful to 410M,1B,1.4B, and 2.8B models. Steps 24k and 73k were also noticeably harmful, yet interestingly that didn't happen for models with 160M parameters or less, meaning that the larger and better the model is the more sensitive it will be towards corrupted data. This works in the opposite way as well: step 55k was relatively helpful for the majority of models

3.3 Optimal parameters combination

Obviously the higher RIG value over the set of probes the better. Despite Pythia (Biderman et al., 2023) suite models using the same architecture, some models were surprisingly bad (31M, 70M) losing to the smallest 14M model in both RIG metrics and official benchmark reports. But the most unexpected thing to notice was the high RIG values on the 410M model. After that, we checked the report again and it occurred that the 410M model outperformed or was on par with 1B, 1.4B, 2.8B, 6.9B, and 12B models on a bunch of general-knowledge domain benchmarks (WSC, LogiQA), though it was under-performing in other (SciQ, ARC Challenge) tasks. What is interesting that the RIG value leads for this model could be noted after 5-10 thousand steps already. SciQ and ARC Challenge could not be properly evaluated by the probe set we used, yet what we noticed, is that parameter combination could be chosen based on a relatively small amount of training steps using a probe set relevant to a task.

3.4 Complexity of the text impact

The main outcome of this experiment is that complex topics, despite they require larger background knowledge, contain less information per token. On average, graduate-level text contained approximately 25-30% less information than 9th-grade one overall models evaluations (Figure 3). Cross entropy value would correlate with Shannon entropy calculations. Our guess based on this observation is that since complex data has lower RIG value, model weight updates and gradients caused by such data would be lower. Hence such data should be utilized in the later stages of model training and presented in larger volumes. Curriculum learning ((Bengio et al., 2009))is not a novel approach in NLP but it was effectively implemented at narrow tasks like knowledge retrieval ((Penha and Hauff, 2020)) or machine translation where there was easy way to structure the data ((Platanios et al., 2019)) based on BLEU score. As for casual language modeling task we believe that RIG value for the sequences in the dataset is an effective parameter to sort them out.

4 Conclusions

Proposed RIG metric is easy to implement and computationally cheap, since it uses only inference pass. Combined with carefully created probes it alters the way we can benchmark models: instead of evaluating model on different tasks with sophisticated evaluation (human reaction, proximity to correct answer, Law of Large Numbers requirement for validation) we could evaluate how much sense does a correct and/or incorrect statement could make to a model. Creating an up to date benchmark is also labor costly task, paired with necessity of either constantly releasing new tests (when existing benchmarks become part of the newer datasets) or making benchmarks private. Our approach on the contrary doesn't require that much labor, it's easily adjusted to a specific task or domain by proper probe creation and it's easy to keep probe set private. RIG value is invaluable addition for curriculum learning applications in NLP, since it finally allows us to sort the data in ascending order of knowledge complexity. It is understandable that RIG value in bits for a fixed statement would be different for different stages of training, yet we might utilize different checkpoints of the same model to re-arrange the dataset in a specific order. That's a main direction for the further

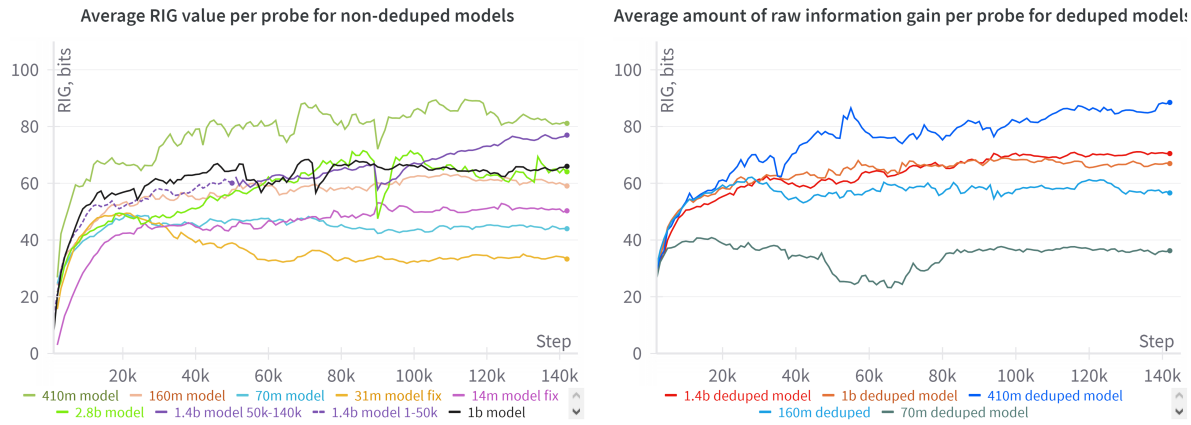

Figure 2: RIG value in bits is averaged over all short probes for all the models. Only truthful probes are counted for this evaluation.

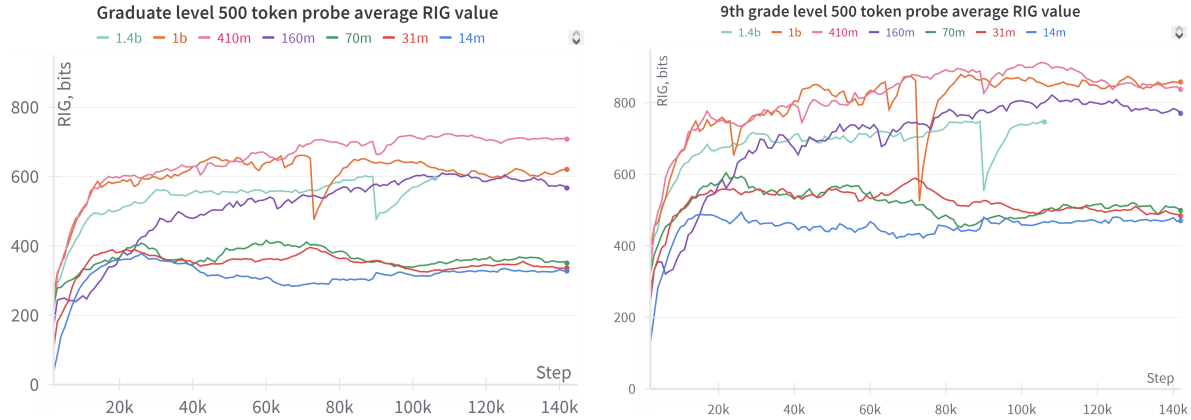

Figure 3: Average RIG value over 500 tokens domain-specific probes. Comparison between 9th grade and graduate level probes related to the same topics

research. As well RIG metrics can help us with mechanistic interpretability of language models if we track how RIG changes over different attention blocks. With some clever probe engineering we could leverage our understanding of how knowledge is processed inside the model. As for search of efficient architectures and parameter combinations and/or solving scaling tasks RIG evaluation is also useful since it helps to understand how many bits of information a model can understand from dataset samples. It also helps us to understand if further training of the model helps or harms us to achieve a certain task, since RIG is more informative than training or validation loss or accuracy. More over, a broad spectrum of domain or task-specific probes could be implemented. That makes the training process interpretative and controllable. Also it's very important to find optimum-parameter settings for small models designed to run on-device and RIG metrics is very useful in this case. Having access to a model of high-quality we can use its RIG evaluations of a dataset to filter it out of harmful or outdated data. RIG evaluation of control probes i.e. *"Charles III is the current monarch of Great Britain"/"Queen Elizabeth II is the current monarch of Great Britain*" could help us analyze, how efficiently model was actualized and if it requires some additional fine-tuning to remove outdated or incorrect information from it's inner knowledge. Finally, we can estimate amount of information contained in articles, papers or books using some high-quality models as subjective experts, especially taking into account that we have access these days to models with incredibly large context windows. The question "How much actual knowledge is there in the text?" could be finally answered using an informative metric operating understandable **bits**.

4.1 Future Work

The main follow up is evaluation of RIG metrics on probes of different complexity on large state of the art models (8B+ parameters) and checking those values corresponding reported benchmarks to make sure that RIG has proper predictive power on model efficiency evaluation. Data filtering and rearranging are another important step to make toward Curriculum Learning general strategy development. Baseline is existing PiLE dataset combined with Pythia suite, allowing to train the models in a specific order instead of simple shuffling the dataset. Existing literature or web-data evaluation on long context model is also an interesting experiment to be done. That's a step towards semi-supervision of data existing in the open domain before it gets mined and implemented in text datasets

Limitations

This work doesn't relate to any specific data used outside of public domain and could be applied towards any encoder-decoder or decoder only language model disregarding nuances of architecture or tandems with expert or reward models implemented. Yet the main requirement to use RIG evaluation is to have access either to logits or token probability distributions for each position in the sequence. Thus closed architecture large models accessible only over API or some web interface could be prevented from evaluation completely. So proprietary solutions like Gemini or GPT-4o could not be evaluated outside. As we change our point of view toward what information is contained in a message, we could expand RIG approach from LLM to multimodal models, yet it is unclear how to interact with computer vision models in this case. As for audio processing and text-to-speech or speech-to-text models RIG seems absolutely reasonable metrics to implement Since a batch of probes is not necessarily as heavy as a batch of training sequences our approach does not create additional hardware requirements. As long as an inference pass of a single sequence could be executed RIG might be evaluated. Yet it should be taken into account, that if RIG is used to control the flow of the training process checkpoint storage is a serious limitation, especially if the model is large. To have an ability to rollback after a bad batch of data you should provide excessive storage capacity. As for evaluation of existing models' checkpoints - uplink speed was the main limiter during the research. Checkpoint download time was always longer than the actual evaluation time interval.

References

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.

Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien, Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward Raff, et al. 2023. Pythia: A suite for analyzing large language models across training and scaling. In *International Conference on Machine Learning*, pages 2397–2430. PMLR.

Nieves Brisaboa, Antonio Farina, Gonzalo Navarro, and José Paramá. 2010. Dynamic lightweight text compression. *ACM Transactions on Information Systems (TOIS)*, 28(3):1–32.

Bruce Croft and John Lafferty. 2003. *Language modeling for information retrieval*, volume 13. Springer Science & Business Media.

Grégoire Delétang, Anian Ruoss, Paul-Ambroise Duquenne, Elliot Catt, Tim Genewein, Christopher Mattern, Jordi Grau-Moya, Li Kevin Wenliang, Matthew Aitchison, Laurent Orseau, et al. 2023. Language modeling is compression. *arXiv preprint arXiv:2309.10668*.

Eugene Galanter. 1962. The direct measurement of utility and subjective probability. *The American journal of psychology*, 75(2):208–220.

Leo Gao, Stella Biderman, Sid Black, Laurence Golding, Travis Hoppe, Charles Foster, Jason Phang, Horace He, Anish Thite, Noa Nabeshima, et al. 2020. The pile: An 800gb dataset of diverse text for language modeling. *arXiv preprint arXiv:2101.00027*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Sagar Imambi, Kolla Bhanu Prakash, and GR Kanagachidambaresan. 2021. Pytorch. *Programming with TensorFlow: Solution for Edge Computing Applications*, pages 87–104.

Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu,

Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Gustavo Penha and Claudia Hauff. 2020. Curriculum learning strategies for ir: An empirical study on conversation response ranking. In *Advances in Information Retrieval: 42nd European Conference on IR Research, ECIR 2020, Lisbon, Portugal, April 14–17, 2020, Proceedings, Part I 42*, pages 699–713. Springer.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom M Mitchell. 2019. Competence-based curriculum learning for neural machine translation. *arXiv preprint arXiv:1903.09848*.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Claude Elwood Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*, 27(3):379–423.