For Distillation, Tokens Are Not All You Need

Mrigank Raman[*]
Machine Learning Department
Carnegie Mellon University
mrigankr@cmu.edu

Pranav Mani
Abridge AI
pranav@abridge.com

Davis Liang
Abridge AI
davis@abridge.com

Zachary C. Lipton
Abridge AI
zack@abridge.com

Abstract

The unwieldy size of state-of-the-art language models presents significant obstacles for deployment, driving up cost and latency. While prior works have offered methods for distilling these larger language models into smaller students, the best previous method is somewhat complex, relying on an RL-based optimization. In this work, we introduce SLIM (**S**parse **L**ogit **I**nfused **M**odeling), a simple method for distilling LLMs that leverages not only samples from the teacher LLM but also the values of the logits produced at each decoding step. Our distillation method uses only the top-5% highest logits along with a dynamic weighting scheme that assigns weights to the KL divergence and cross-entropy loss based on the relative confidence between the student and teacher models. Our experiments demonstrate that SLIM produces models that are better at a wide range of downstream NLP tasks compared to supervised finetuning, vanilla knowledge distillation, and the recently proposed MiniLLM. Contrary to other methods, our method is scalable to much larger teacher ($\sim$ 70B parameters). We also provide an intuition for the superior performance of SLIM via established sample complexity bounds within simplified scenarios.

1 Introduction

Recent work in large language modeling has demonstrated that increasing the number of parameters and overall model size [OpenAI, 2023, Brown et al., 2020] results in significantly improved generalization across a diverse set of tasks [Bubeck et al., 2023]. State-of-the-art large language models (LLMs), such as PaLM [Chowdhery et al., 2022], ChatGPT [OpenAI, 2023], and Claude, have grown so large that experimenting with them has become impractical for many researchers in both academia and industry alike.

In tandem, the impressive quality of responses produced by LLMs have led to their widespread application to data annotation [Wang et al., 2021]. Along these lines, methods like instruction tuning [Ouyang et al., 2022] and Self-Instruct [Wang et al., 2022a] have resulted in successful instruction-tuned models such as InstructGPT [Ouyang et al., 2022] and Alpaca [Taori et al., 2023]. In particular, Ouyang et al. [2022] showed that a 1.3B parameter model, after instruction tuning, can outperform its 175B parameter GPT-3 counterpart.

Yet, despite the surge in popularity of synthetically labeled corpora and instruction tuning, methods have not evolved beyond the basics – practitioners typically perform supervised finetuning (SFT) on

[*] Work done as an intern at Abridge Inc.

Workshop on Instruction Tuning and Instruction Following at NeurIPS 2023.

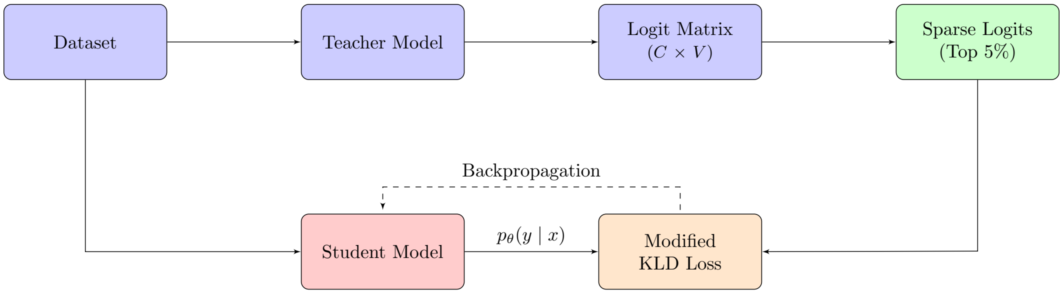

Figure 1: An illustration of our method. First, for a sequence of length C from the dataset, we calculate the logits using the teacher model. Next, we retain the top 5% of these logits and save them in a sparse form on disk. During training, we calculate a modified KLD loss. This loss is based on both the stored sparse logits from the teacher and the outputs from the student model. We then use this loss to update the student model.

raw generated text, or hard labels, while hoping to learn the conditional distributions characteristic of the original LLMs that were used to generate these labels. However, it's clear that each hard label, or individual token, provides only a shallow glimpse of the underlying distribution it was sampled from. One can only hope that over a large number of samples, we can approximate this underlying distribution. By leveraging the model's output distribution over the entire vocabulary at each time-step as in knowledge distillation (KD) [Hinton et al., 2015], we uncover information equivalent to many samples from the underlying latent distribution.

While work in the area of knowledge distillation is well-studied for small (<1B parameter) models [Sanh et al., 2019] [Wang et al., 2020], the application of KD to LLMs has several known issues. One particular issue arises when the student model overestimates the low-probability regions of the teacher distribution while being insufficiently expressive to cover all modes of the teacher model [Ji et al., 2023]. Recent work like MiniLLM [Gu et al., 2023], proposes replacing the forward Kullback-Leibler divergence (KLD) objective in the standard KD approach with reverse KLD to prevent the student model from overestimating the low-probability regions of the teacher distribution. Then, the authors derive a reinforcement learning based optimization procedure to learn this objective. In this paper, we strip away these layers of complexity and provide an approach that is not only significantly simpler and more scalable to larger teacher models, but also results in a student model that outperforms both MiniLLM and SFT on multiple evaluation sets.

Overall, our contributions are thus,

- We introduce SLIM (**S**parse **L**ogit **I**nfused **M**odeling), a method that utilizes the top 5% of logits from an open-source Large Language Model (LLM) to create a dataset of logit values. This compact dataset is subsequently employed to train models through knowledge distillation using a modified KL divergence loss.
- Our experiments reveal that SLIM effectively boosts LLM capabilities in both instruction following and multiple downstream tasks.
- We provide an intuition for the superior performance of SLIM via established sample complexity bounds within simplified scenarios. These bounds offer valuable insights into the factors contributing to the success of our approach, enhancing our understanding of instruction tuning processes.

2 Method

Notation We denote the timestep in a sequence with subscript and sample in a dataset with superscript. Thus, n sequences are denoted as $\{(x_m)^i_{m>0}\}^n_{i=1}$. For ease of writing, where convenient, we drop the superscript when considering any arbitrary sample sequence. Accordingly, x_m is the m-th element from an arbitrary sequence sample. Further, we sometimes denote a sequence $x_1, x_2, \ldots, x_{i-1}$ using $\mathbf{x_i}$. For labels, we use y_i to represent the correct label at timestep i in an arbitrary sequence sample. For instance, $(\mathbf{x_i}, y_i)$ represents an arbitrary datapoint of input sequence

$x_1, x_2, \ldots, x_{i-1}$ and the correct completion label y_i. We use $\mathbf{y}_i$ to represent a one-hot vector over the vocabulary representing y_i.

We are focused on distilling from commercial LLMs to student language models with parameter counts in the range of $70M - 13B$. Our dataset consists of n sequences $\{(x_m)^i_{m>0}\}^n_{i=1}$. Our objective is to approximate $p(x_m \mid x_1 x_2 \cdots x_{m-1})$. Conventionally, this is done by modeling the distribution as $p_\theta(x_m \mid x_1 x_2 \cdots x_{m-1})$, where θ represents the learnable parameters. θ is then obtained by minimizing the Causal Language Modeling (CLM) loss, defined for a sample as:

$$\mathcal{L}_{CLM} = -\frac{1}{N} \sum_{i=1}^{N} \log p_\theta(x_i \mid x_1, x_2, \ldots, x_{i-1}) \tag{1}$$

where N is the length of the sequence, and y_i and $x_1, x_2, \ldots, x_{i-1}$ are the actual and preceding tokens, respectively. It is useful to note that (i) the model parameters θ are shared across all time steps to achieve better generalization for longer sequences while maintaining a manageable model size; (ii) each term in the CLM loss includes the KL-divergence between the predicted distribution and the one hot distribution of each ground truth sample. When several samples are obtained for the same input location, the terms in the CLM loss corresponding to these samples begin to resemble the KL-divergence between the predicted distribution and the true ground truth distribution. In contrast to this traditional method, our approach leverages logits from a larger and more capable model. Our methodology (Figure 1) consists of two primary components: first, a logit dataset creation pipeline, and second, training our consumer-grade model using knowledge distillation [Hinton et al., 2015] with a modified KL-divergence loss.

2.1 Logit Dataset Creation

In the context of our dataset, we introduce a strategy that leverages a larger and more proficient teacher model to generate a supplementary dataset comprising logits. These logits are calculated by applying the teacher model on the same data points as in our original dataset. The idea is to exploit the intricate relationships captured by the teacher model, and to use these as soft targets for training our consumer-grade model. To elaborate, for any given example in the dataset with a context length of C, and considering a vocabulary size of V, the resulting logit matrix would naturally assume dimensions of $C \times V$.

Storing such extensive logit matrices for every example in the dataset could potentially lead to significant memory overhead, especially if the intent is to persistently store these logits on disk for future use. As a remedy for this scalability issue, our proposal involves a trade-off between granularity and computational efficiency. Specifically, we retain only the top 5% of logits for each token within the example, while zeroing out all other entries in the logit matrix. This strategic pruning enables us to store the complete logit matrix in a sparse representation, dramatically reducing the memory footprint.

Our empirical analysis substantiates the efficacy of this approach. In Section 3, we present empirical evidence showing that, despite the sparsity introduced, our logit-based distillation method consistently outperforms traditional supervised fine-tuning methods across various benchmarks. This serves as a validation of the effectiveness of our approach in balancing the need for a rich feature set with computational and storage constraints.

2.2 Training via knowledge distillation

Consider a dataset, denoted as $\mathcal{D}$, that comprises n sequences. Correspondingly, we have constructed a logit dataset, symbolized as $\mathcal{T}$. With these resources at hand, our objective is to fine-tune our model using the principles of knowledge distillation. This entails a process wherein the knowledge encapsulated in a more complex, often pre-trained model (the teacher) is distilled into a simpler model (the student). To achieve this, we employ a modified KL-divergence loss.

For every datapoint $(\mathbf{x}_i, y_i) \in \mathcal{D}$ and its associated logit $\mathbf{t_i} \in \mathcal{T}$ (vector of length equal to the vocabulary), the input $\mathbf{x}_i$ is processed through our student model to produce logits $\mathbf{s}_i$. To regulate the influence of these logits, a temperature parameter τ is introduced, leading to normalized logits $\mathbf{s}_i^\tau = \mathbf{s}_i/\tau$ and $\mathbf{t}_i^\tau = \mathbf{t}_i/\tau$.

We can delineate our loss computation as:

- Cross-Entropy Loss:
$$\mathcal{L}_{ce} = -\mathbf{y}_i^T \cdot \log(\mathbf{s}_i) \tag{2}$$
- KD Loss:
$$\mathcal{L}_{kd} = -(\mathbf{t}_i^\tau)^T \cdot \log(\mathbf{s}_i^\tau) \tag{3}$$

Our final loss formulation draws inspiration from the work of Zhou et al. [2021]. We inform the weight given to the KD loss by factoring in the ratios of the true/teacher logit corresponding to the hard label against the student logit for the hard label. We could express the teacher logit for the hard label as $t_i = \mathbf{y}_i^T \cdot \mathbf{t}_i$ and likewise the student logit as $s_i = \mathbf{y}_i^T \cdot \mathbf{s}_i$. We then compute our loss as:

$$\mathcal{L}_{final} = \mathcal{L}_{ce} + \alpha \left(1 - \exp\left(-\frac{s_i}{t_i}\right)\right) \mathcal{L}_{kd} \tag{4}$$

Here, α is a tunable hyperparameter. Note that this loss is adaptive by nature: if the teacher (represented by logits from $\mathcal{T}$) presents higher confidence for a given sample, the influence of the KD loss is amplified. Conversely, if the student model is more confident in its prediction, the weight on the KD loss is attenuated. This weighting strategy allows our student model to learn from the teacher while being robusts to contexts where the teacher model is subpar.

3 Experiments

3.1 Instruction Following

To understand the effectiveness of our approach, we initially focus on the instruction-following task. In this context, we use the train and val splits of Dolly dataset constructed by Gu et al. [2023]. An important distinction in our methodology, as compared to that of Gu et al. [2023], is our decision to forego the use of $\mathcal{D}_{pt}$ for pretraining.

We evaluate the trained models on 3 different instruction-following datasets:

- **DollyEval:** A test split of Dolly dataset consisting of 500 examples created by Gu et al. [2023].
- **VicunaEval** [Xu et al., 2023a]: The 80 challenging questions used in the evaluation of the Vicuna model.
- **SelfInst** [Wang et al., 2022a]: A user-oriented instruction-following set with 252 samples.

We use Rouge-L [Lin, 2004] score and GPT-4 [OpenAI, 2023] feedback to evaluate the instruction following capability of the models. Wang et al. [2022b] demonstrated that Rouge-L is suitable for large-scale instruction-following evaluation. In addition to reporting Rouge-L scores, we also leverage GPT-4 to provide a feedback score similar to Gu et al. [2023] in the range of 1-10 and then report the ratio of the average GPT-4 score of the model generations to that of the ground truth. We use supervised finetuning using hard labels (SFT) and MiniLLM as our baselines and use the 7B models as the backbone for training. We don't compare SLIM against vanilla KD since it is computationally infeasible with a 30B parameter teacher model.

From our experiments on three distinct datasets and model versions, as presented in Table 1, our approach consistently outperforms the supervised finetuning using hard labels (SFT) across all metrics, datasets and model types while also outperforming MiniLLM across all dataset using the LLaMA model. Notably, in the LLaMA 2 and MPT experiments, MiniLLM results are absent as the authors did not release their models for these versions.

3.2 Downstream Tasks

Beyond merely examining our method's proficiency in instruction-following tasks, we extended our evaluation to encompass a variety of downstream tasks. For this assessment, we train our models using the open assistant Guanaco dataset introduced by Dettmers et al. [2023].

Post-training, we test the models across four distinct tasks in the 5-shot setting:

	LLaMA					
	DollyEval		VicunaEval		SelfInst	
	Rouge-L	GPT-4	Rouge-L	GPT-4	Rouge-L	GPT-4
Teacher (LLaMA 13B)	29.7	85.3	19.4	66.3	23.4	76.2
SFT	26.3	79.47	17.5	61.5	20.8	70.6
MiniLLM	28.7	82	19.8	63.8	20.5	72.1
SLIM (Ours)	**29.2**	**82.6**	**20.1**	**64.1**	**23.2**	**73.2**
	LLaMA 2					
	DollyEval		VicunaEval		SelfInst	
	Rouge-L	GPT-4	Rouge-L	GPT-4	Rouge-L	GPT-4
Teacher (LLaMA 2 13B)	30.2	88.9	21.3	69.3	25.1	79.1
SFT	26.5	80.3	18.3	62.8	21.3	73.4
SLIM (Ours)	**29.3**	**84.6**	**19.9**	**67.1**	**23.4**	**75.1**
	MPT					
	DollyEval		VicunaEval		SelfInst	
	Rouge-L	GPT-4	Rouge-L	GPT-4	Rouge-L	GPT-4
Teacher (MPT-30B-instruct)	44.0	94.7	19.3	67.3	23.5	76.1
SFT	28.6	79.5	16.62	60.7	19.9	70.7
SLIM (Ours)	**31.1**	**83.3**	**17.83**	**63.4**	**22.9**	**73.8**

Table 1: We report the Rouge-L and GPT-4 agreement scores on 3 different datasets across 3 different models. We do not have MiniLLM numbers for LLaMA 2 and MPT experiments since the authors did not open-source their models with these backbones.

- **ARC** [Clark et al., 2018]: A challenge focused on reasoning-based question answering.
- **Hellaswag** [Zellers et al., 2019]: A task-centered on predicting the most likely ending to a given situation.
- **MMLU** [Hendrycks et al., 2020]: Designed to evaluate models on multiple-choice questions.
- **TruthfulQA** [Lin et al., 2021]: A dataset that emphasizes truthful answering capabilities.

For comparative evaluation, we use SFT as our reference benchmark. We don't compare SLIM against vanilla KD since it is computationally infeasible with a 70B parameter teacher model. In our setup, the LLaMA 2 70B served as the teacher model. Notably, our method exhibited a marked improvement in the performance of a 7B model, reflecting an average improvement of 1% across the four mentioned tasks. Especially significant were the performance boosts observed for MMLU and Hellaswag, each surpassing the 1% mark (as detailed in Table 2). A parallel trend was observed when employing a more sizeable 13B model, as evidenced in Table 3.

	SFT	SLIM (Ours)
MMLU	46.3	47.6
Hellaswag	79.6	81.3
ARC	54.2	54.2
Truthful_qa	43.9	44.9
Average	56.0	57.0

Table 2: Performance of Llama2-7B model on various downstream tasks when finetuned using the open assistant guanaco dataset. Since MMLU is an ensemble of 57 tasks, we believe that 1.3% is a sizeable boost in performance.

	SFT	SLIM (Ours)
MMLU	52.4	54.6
Hellaswag	82.5	83.8
ARC	59.6	60.2
Truthful_qa	43.9	44.6
Average	59.6	60.8

Table 3: Performance of Llama2-13B model on various downstream tasks when finetuned using the open assistant guanaco dataset. Since MMLU is an ensemble of 57 tasks, we believe that 2.2% is a sizeable boost in performance.

3.3 Pretraining

Recent work by Li et al. [2023] demonstrated the value of generating synthetic data using larger, more capable teacher LLMs to train smaller, consumer-grade student LLMs. In light of this effective method of data generation, we decided to analyze the sample efficiency of our method on such synthetic datasets. We additionally compare our method against SFT and vanilla knowledge distillation.

Specifically, we leverage a pretrained Pythia-6.9B model and sample a synthetic dataset of 10000 generations using a temperature of 1.0 without any seed context. Along with these generations, we also extract the top 5% of logits for each token in each generated sequence. We then train a randomly initialized Pythia-160M model for 100 epochs on various subsamples of the dataset (from 625 to 10000 examples in each subsampling).

Overall, we show in 2 that our method is consistently more sample efficient than both vanilla KD and SFT across all sizes of subsamples.

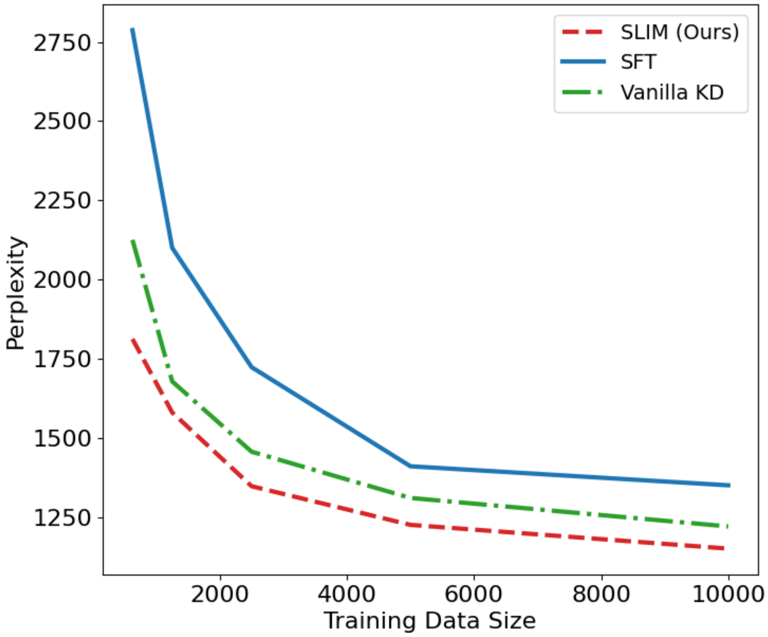

Figure 2: Perplexity of a Pythia-160M model when trained using different data sizes of a synthetic dataset generated using the Pythia-6.9B model. It is clear that SLIM requires less than half as many samples to achieve the same perplexity as SFT while being slightly better than vanilla KD as well.

4 Analysis

In this section, we will formalize the intuition on the benefits of using whitebox KD over vanilla supervised finetuning from a sample complexity perspective in a simplified setup. We make the following assumptions:

A.1 We have access to a vocabulary $\mathcal{V}$ of tokens which has a cardinality of V.

A.2 Given an input sequence $s \in \mathcal{S}$, we have access to $\mathbf{x_s}$ which is a d-dimensional embedding vector corresponding to s

A.3 Existence of ground truth distribution: Given an input sequence embedding $\mathbf{x_s}$, there exists a distribution $p(y \mid \mathbf{x_s})$ such that the ground truth token $t \sim p(y \mid \mathbf{x_s})$

A.4 Existence of a perfect teacher: Given an input sequence embedding $\mathbf{x_s}$, there exists a function $q(\mathbf{x})$ such that $softmax(q(\mathbf{x_s})) = p(y \mid \mathbf{x_s})$. We call $q(\mathbf{x})$ as the teacher.

A.5 Linear Assumption: Our goal is to learn $\theta^* \in \mathbb{R}^{d \times V}$ such that $\theta^* \in argmin\ \mathbb{E}\|\theta^T \mathbf{x_s} - q(\mathbf{x_s})\|_2^2$

Assumptions A.1–A.3 are generally benign. While A.4 may not apply to a wider class of problems, it is a common assumption for the distillation problem class. A.5 is a strong assumption that qualifies our setup as simplistic.

Proposition 1 *(adapted from [Haussler, 1992]) With high probability, $\|\hat{\theta^*} - \theta^*\| < \epsilon$ holds when $n > \mathcal{O}(\frac{1}{\epsilon^2})$ and the optimization algorithm is SFT, where $\hat{\theta^*}$ is derived using Empirical Risk Minimization.*

We refer readers to [Haussler, 1992] for the original statement and proof. The idea in adapting to our work is that, for SFT, we do not use the teacher and directly rely on the ground truth next token t for a given sequence s. In this setup, predicting the next token can be approached as a classification problem, with $\mathcal{V}$ representing the set of potential classes. Given that the teacher might not always be linear, the agnostic PAC learning theory [Haussler, 1992] indicates the need for $\mathcal{O}(\frac{1}{\epsilon^2})$ samples to derive an estimator that is within ϵ distance of the optimal linear estimator.

Proposition 2 *With high probability, $\|\hat{\theta^*} - \theta^*\| < \epsilon$ holds when $n > \mathcal{O}(\frac{1}{\epsilon})$, given that $\hat{\theta^*}$ is obtained via SGD on the function $h(\theta) = \mathbb{E}[\|\theta^T \mathbf{x_s} - q(\mathbf{x_s})\|_2^2]$ in the context of whitebox KD.*

With knowledge distillation, it is presumed that we can access $q(\mathbf{x})$. This allows us to apply SGD to the function $h(\theta)$, which is inherently a strongly convex function in θ. Building on the work of Lacoste-Julien et al. [2012], we deduce that knowledge distillation necessitates $\mathcal{O}(\frac{1}{\epsilon})$ samples to learn an estimator at an ϵ distance from the best linear estimator. This insight underscores the efficacy of knowledge distillation, especially in straightforward scenarios. Our experiments confirm that even for Large Language Models, knowledge distillation consistently delivers better results. The sample efficiency of knowledge distillation and our method is evident from Figure 2.

5 Prior Work

Instruction Tuning The paradigm of instruction tuning was introduced by Ouyang et al. [2022] and they demonstrated that smaller models could perform at par with 100x larger models using instruction tuning. Since then, there have been a lot of work on improving instruction tuning by curating high quality datasets [Taori et al., 2023, Han and Tsvetkov, 2022, Lee et al., 2023, Zhou et al., 2023] and improving the efficiency of finetuning [Dettmers et al., 2023, Xu et al., 2023b].

Knowledge Distillation Knowledge distillation was first proposed by Hinton et al. [2015]. Since its introduction, it has been widely adopted in Computer Vision [Chen et al., 2017, Zhang et al., 2019, 2020, Xu et al., 2020] and Natural Language Processing [Liu et al., 2019, Haidar and Rezagholizadeh, 2019, Arora et al., 2019]. While there have been studies on knowledge distillation for small-scale language models [Sanh et al., 2019, Wu et al., 2021], there are very few works [Gu et al., 2023] that address knowledge distillation for Large Language Models. Ma et al. [2021] explores a similar concept in computer vision by keeping the top 10% of logits for knowledge distillation. Their results on CIFAR-10 and Tiny-ImageNet show that this method results in a 20% drop compared with training from scratch. We hypothesize that causal language modeling, unlike image classification, contains classes (tokens) that follow a Zipfian distribution and a overwhelming majority of the probability mass is contained within the top 5% of tokens.

6 Conclusion

In this work we introduce SLIM, a simple augmentation to the classic knowledge distillation recipe to produce high quality instruction tuned Large Language Models. SLIM both prevents the student model from overestimating the low-probabilty regions of the teacher distribution while also producing a compact yet substantial dataset of sparse logit and raw text sequence pairs that can be easily shared – especially in resource constrained cases where performing inference using the original teacher model is difficult. Our extensive experiments demonstrate the efficacy of SLIM as compared with standard instruction tuning, vanilla knowledge distillation, and competing distillation methods like MiniLLM on a range of downstream NLP tasks.

References

OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. URL https://api.semanticscholar.org/CorpusID:257532815.

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

Sébastien Bubeck, Varun Chandrasekaran, Ronen Eldan, Johannes Gehrke, Eric Horvitz, Ece Kamar, Peter Lee, Yin Tat Lee, Yuanzhi Li, Scott Lundberg, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. *arXiv preprint arXiv:2303.12712*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *arXiv preprint arXiv:2204.02311*, 2022.

Shuohang Wang, Yang Liu, Yichong Xu, Chenguang Zhu, and Michael Zeng. Want to reduce labeling cost? GPT-3 can help. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4195–4205, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-emnlp.354. URL https://aclanthology.org/2021.findings-emnlp.354.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35: 27730–27744, 2022.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022a.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

Wenhui Wang, Furu Wei, Li Dong, Hangbo Bao, Nan Yang, and Ming Zhou. Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. *Advances in Neural Information Processing Systems*, 33:5776–5788, 2020.

Haozhe Ji, Pei Ke, Zhipeng Hu, Rongsheng Zhang, and Minlie Huang. Tailoring language generation models under total variation distance. *arXiv preprint arXiv:2302.13344*, 2023.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. Knowledge distillation of large language models. *arXiv preprint arXiv:2306.08543*, 2023.

Helong Zhou, Liangchen Song, Jiajie Chen, Ye Zhou, Guoli Wang, Junsong Yuan, and Qian Zhang. Rethinking soft labels for knowledge distillation: A bias–variance tradeoff perspective. In *International Conference on Learning Representations*, 2021. URL `https://openreview.net/forum?id=gIHd-5X324`.

Canwen Xu, Daya Guo, Nan Duan, and Julian McAuley. Baize: An open-source chat model with parameter-efficient tuning on self-chat data. *arXiv preprint arXiv:2304.01196*, 2023a.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics. URL `https://aclanthology.org/W04-1013`.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates, December 2022b. Association for Computational Linguistics. doi: 10.18653/v1/2022.emnlp-main.340. URL `https://aclanthology.org/2022.emnlp-main.340`.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*, 2023.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. *ArXiv*, abs/1803.05457, 2018. URL `https://api.semanticscholar.org/CorpusID:3922816`.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. Hellaswag: Can a machine really finish your sentence? *arXiv preprint arXiv:1905.07830*, 2019.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958*, 2021.

Yuanzhi Li, Sébastien Bubeck, Ronen Eldan, Allie Del Giorno, Suriya Gunasekar, and Yin Tat Lee. Textbooks are all you need ii: phi-1.5 technical report. *arXiv preprint arXiv:2309.05463*, 2023.

David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and Computation*, 100(1):78–150, 1992. ISSN 0890-5401. doi: https://doi.org/10.1016/0890-5401(92)90010-D. URL `https://www.sciencedirect.com/science/article/pii/089054019290010D`.

Simon Lacoste-Julien, Mark Schmidt, and Francis Bach. A simpler approach to obtaining an o (1/t) convergence rate for the projected stochastic subgradient method. *arXiv preprint arXiv:1212.2002*, 2012.

Xiaochuang Han and Yulia Tsvetkov. Orca: Interpreting prompted language models via locating supporting data evidence in the ocean of pretraining data. *arXiv preprint arXiv:2205.12600*, 2022.

Ariel N Lee, Cole J Hunter, and Nataniel Ruiz. Platypus: Quick, cheap, and powerful refinement of llms. *arXiv preprint arXiv:2308.07317*, 2023.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

Yuhui Xu, Lingxi Xie, Xiaotao Gu, Xin Chen, Heng Chang, Hengheng Zhang, Zhensu Chen, Xiaopeng Zhang, and Qi Tian. Qa-lora: Quantization-aware low-rank adaptation of large language models. *arXiv preprint arXiv:2309.14717*, 2023b.

Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning efficient object detection models with knowledge distillation. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/e1e32e235eee1f970470a3a6658dfdd5-Paper.pdf`.

Feng Zhang, Xiatian Zhu, and Mao Ye. Fast human pose estimation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3517–3526, 2019.

Linfeng Zhang, Yukang Shi, Zuoqiang Shi, Kaisheng Ma, and Chenglong Bao. Task-oriented feature distillation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 14759–14771. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/a96b65a721e561e1e3de768ac819ffbb-Paper.pdf`.

Kunran Xu, Lai Rui, Yishi Li, and Lin Gu. Feature normalized knowledge distillation for image classification. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV*, page 664–680, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58594-5. doi: 10.1007/978-3-030-58595-2_40. URL `https://doi.org/10.1007/978-3-030-58595-2_40`.

Jian Liu, Yubo Chen, and Kang Liu. Exploiting the ground-truth: An adversarial imitation based knowledge distillation approach for event detection. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):6754–6761, Jul. 2019. doi: 10.1609/aaai.v33i01.33016754. URL `https://ojs.aaai.org/index.php/AAAI/article/view/4649`.

Md Akmal Haidar and Mehdi Rezagholizadeh. Textkd-gan: Text generation using knowledge distillation and generative adversarial networks. In *Advances in Artificial Intelligence: 32nd Canadian Conference on Artificial Intelligence, Canadian AI 2019, Kingston, ON, Canada, May 28–31, 2019, Proceedings 32*, pages 107–118. Springer, 2019.

Siddhartha Arora, Mitesh M. Khapra, and Harish G. Ramaswamy. On knowledge distillation from complex networks for response prediction. 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 3813–3822, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1382. URL `https://aclanthology.org/N19-1382`.

Zhengxuan Wu, Atticus Geiger, Josh Rozner, Elisa Kreiss, Hanson Lu, Thomas Icard, Christopher Potts, and Noah D Goodman. Causal distillation for language models. *arXiv preprint arXiv:2112.02505*, 2021.

Haoyu Ma, Yifan Huang, Tianlong Chen, Hao Tang, Chenyu You, Zhangyang Wang, and Xiaohui Xie. Stingy teacher: Sparse logits suffice to fail knowledge distillation. 2021.