Improving Training Methods to Enhance the Acceptance Rate of a Draft Model on EAGLE

Jungmin Ha[1], Jaehyeok Choi[1] and Jungwook Choi[1]

[1] Department of Electronic Engineering, Hanyang University, Seoul, South Korea, {mechanismha, wjdrmfthsus, choij}@hanyang.ac.kr

Abstract

This paper presents methods for improving the training of draft models within the EAGLE speculative decoding framework, particularly when the original dataset used to train the base model is unavailable. Using the Phi-3-medium-4k-instruct model, we show that pretraining the draft model on the base model's benchmark datasets, followed by instruction fine-tuning, significantly improves the acceptance rate. Pretraining allowed the draft model to acquire general knowledge, avoiding overfitting to instructions. Instruction fine-tuning aligned the draft model with the base model's instruction-tuned characteristics. This approach addresses dataset unavailability while extending the EAGLE framework's applicability to a broader range of instruction-tuned models.

Keywords— *Deep Learning, Machine Learning, Speculative Decoding, Draft Model, Instruction Tuning*

I. Introduction

In recent years, the rapid growth of Large Language Models (LLMs) has led to increased computational demands. LLMs generate text one token at a time, with each token's prediction depending on all previous tokens. This sequential nature creates a bottleneck, especially for long sequences. Various methods have been developed to accelerate autoregressive inference, with Speculative Decoding standing out for its ability to improve LLM latency. Specifically, the EAGLE (Extrapolation Algorithm for Greater Language-model Efficiency) framework [3] is a leading speculative decoding algorithm, leveraging both token and feature sequences during drafting. However, training a draft model for EAGLE poses challenges. Our research aims to address these challenges by introducing novel methods to enhance draft model acceptance rates. The work in this paper builds on the insights from participation in the Samsung Computer Engineering Challenge 2024. **Terminologies.** In this paper, the term **"Base model"** refers to the large language model that is targeted for acceleration. **"Draft model"** is the model utilized for draft generation. The term **"Feature"** denotes the last hidden state preceding the language model head. **"Draft Length"** refers to the number of tokens generated by the draft model in a single iteration. **"Acceptance Rate"** is the ratio of accepted tokens generated by the Draft Model to the total Draft Length.

II. Related Works

A. Speculative Decoding

Speculative decoding is a method that speeds up LLM inference by using a lighter draft model to suggest multiple token predictions [1, 2]. These token candidates are then verified by the base model, enabling parallel processing and faster inference without sacrificing accuracy. If the draft model's tokens align with the base model, they're accepted; otherwise, the base model corrects them. The primary goal is to minimize the need for the base model to perform autoregressive decoding, thereby improving the speed of inference. The effectiveness of this approach depends on the draft model's acceptance rate—higher rates lead to more tokens being accepted without additional computation. Thus, aligning the draft model with the base model is key to maximizing efficiency.

B. EAGLE Framework

EAGLE's draft model exploits shifted token sequences to resolve feature-level autoregression uncertainty, achieving high accuracy in the drafting phase [3]. Instead of training directly on tokens, the EAGLE draft model learns from features of the base model's output. Since datasets are in text form, they must be transformed into features by passing them through the base model. For fixed datasets, both prompts and labels are forwarded through the base model. For generated datasets, only prompts are used, and the base model generates responses autoregressively. Although the ablation study in [3] suggests the EAGLE draft model's

low sensitivity to its training dataset (whether it is fixed or generated dataset), our research highlights that a significant mismatch between a fixed dataset and the base model can lead to a more pronounced impact on performance.

III. Training Methods

This section addresses the challenges of training a draft model for a base model with an unknown original dataset. Our approach involved pretraining and instruction tuning using open-source benchmark datasets. We utilized the EAGLE-1 speculative decoding framework, with Phi-3-medium-4k-instruct (Phi-3) as the base model.

A. Ideal Training of a Draft Model

The draft model's goal is to mimic the base model's token generation, but training on the base model's outputs is inefficient. A well-representative fixed dataset can yield similar results without autoregressive generation [3]. In prior research [3], the draft models of Vicuna (7B, 13B, 33B) and LLama2-Chat (7B, 13B, 70B) were trained on the ShareGPT dataset. Notably, Vicuna models were directly fine-tuned on the same dataset from LLama2, and LLama2-Chat models were also adapted to similar conversation-based datasets. Therefore, these draft models were easier to train as the open-source ShareGPT dataset was readily available for alignment with the base model. To examine this, we trained and reproduced the draft model for Vicuna-7B-v1.3 on the ShareGPT dataset, achieving comparable outcomes (see Table 1). In contrast, the Phi-3 draft model, reproduced from the same ShareGPT dataset, produced poor results on its own benchmark datasets (see Table 4 - **'Rep'**).

Vicuna-reproduced	Vicuna-original
39.36 tokens/s	39.56 tokens/s
(99.5% reproduced)	(100%)

Table 1. Performance comparison of the reproduced draft model of Vicuna and the original EAGLE-Vicuna Draft Model, on the official EAGLE GitHub repository([3]). Vicuna-reproduced was trained on ShareGPT dataset. Generation was tested using a prompt from ShareGPT. The evaluation was conducted on a Jetson AGX Orin 64GB, utilizing the EAGLE-2 framework.

B. Challenges

The underperformance of the reproduced Phi-3 draft model (**'Rep'**) is attributed to the mismatch between the Phi-3 base model and the ShareGPT dataset (Fig. 1). To address this, we generated outputs from the Phi-3 base model using questions from ShareGPT. In contrast to the ablation study in prior research [3], our experiments showed that training the draft model on this autoregressively generated dataset significantly improved benchmark performance (see Table 4 - **'Rep'** & **'Autoreg'**). However, the acceptance rate for the draft model trained on this dataset (**'Autoreg'**) remained below 60%, indicating that the dataset still did not adequately represent the Phi-3 base model. This highlighted the need for access to the Phi-3 model's original dataset. However, the primary challenge we faced was the lack of transparency and accessibility regarding the original dataset.

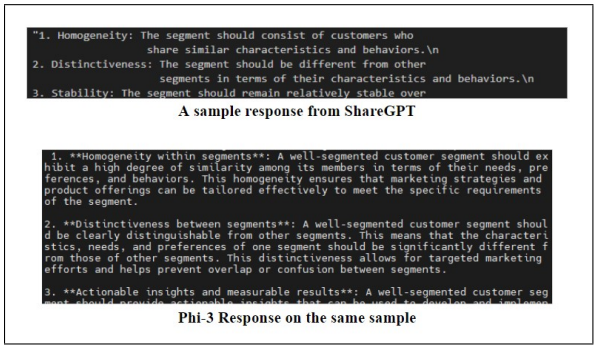

Fig. 1. Mismatch between a sample response from ShareGPT and the actual response of the Phi-3-medium-4k-instruct model from the same prompt.

Train Dataset	Task	Samples
MMLU	Question Answering	100
Big-Bench Hard	Question Answering	250
HellaSwag	Sentence Completion	10,000
ARC Challenge	Question Answering	1,119
ARC Easy	Question Answering	2251
BoolQ	Question Answering	9,427
CommonsenseQA	Question Answering	9741
MedQA	Question Answering	10,178
OpenBookQA	Question Answering	4,957
PIQA	Question Answering	16,113
Social IQA	Question Answering	10,000
WinoGrande	Fill in a Blank	10,000
GSM8K	Mathematical Reasoining (Text Generation)	5,000
MBPP	Code Generation (Text Generation)	374
	Total	89510

Table 2. Distribution of the pretraining dataset randomly sampled from the benchmark dataset of the Phi-3-medium-4k-instruct model.

C. Our Proposed Method

To overcome challenges, we leveraged benchmark datasets where the Phi-3 model demonstrated strong performance. We found that instruction tuning after pretraining on benchmarks produced the best results.

Since the Phi-3 model's original dataset is not publicly available, we collected 89,510 randomly sampled data points from 14 benchmarks of the Phi-3 base model (see Table 2), all of which have distinct training and test datasets. During pretraining, the draft model was trained on this dataset (11-13M tokens) without any augmentation, and the hyperparameters and the optimizer were set according to [3], except for the learning rate, which was optimized through sweeping between 2e-6 and 3e-5. The pretrained draft model (**'Pre'**) demonstrated performance comparable to the model trained on the generated dataset (**'Autoreg'**) (see Table. 4 - **'Pre'**) and even outperformed

it on unseen datasets such as AGIEval, TruthfulQA, and HumanEval, despite not being trained on the actual generated dataset. This suggests that the benchmarks could effectively represent the base model, similar to the generated dataset from ShareGPT.

To further enhance the performance of the pretrained draft model, we fine-tuned it on instruction-augmented datasets. For tasks with multiple-choice formats, instructions were added at the beginning of the prompts (see Fig. 2), while datasets like GSM8K and MBPP, which lack multiple-choice formats, were left unchanged. As explained in this survey [4], these instructions guided the model to produce more predictable outputs, aligning its responses with the dataset labels (see Fig. 3). As a result, the draft model was able to train on features that were well-aligned with the base model output, generated from a single forward pass. (II. - B.). However, pretraining on unpredictable outputs remains crucial to prevent overfitting to tasks with instructions and to preserve the draft model's general alignment with the base model.

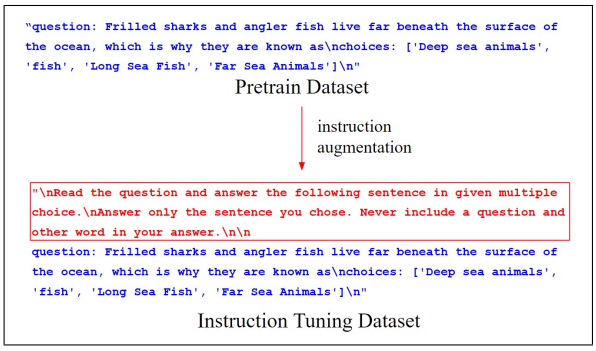

Fig. 2. An example of instruction augmentation.

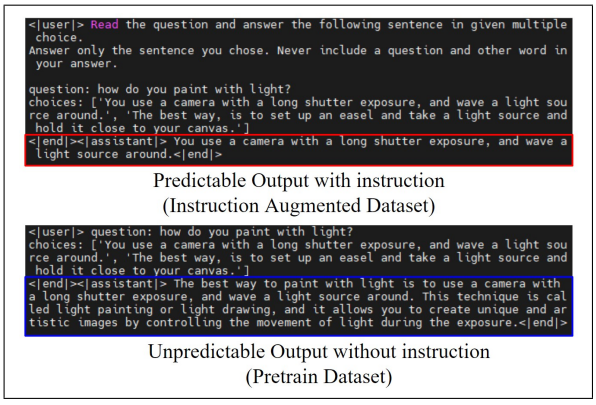

Fig. 3. The outputs of the Phi-3-medium-4k-instruct model on examples of the instruction augmented dataset and the pretrain dataset.

IV. EXPERIMENTS

We trained five different draft models (see Table 4) each on their respective dataset. The best-performing

Datasets	Task	Instruction	Samples
AGI Eval	Question Answering	O	30
MMLU	Question Answering	O	30
BigBench Hard	Question Answering	O	30
HellaSwag	Setence Completion	O	30
ARC Challenge	Question Answering	O	30
ARC Easy	Question Answering	O	30
BoolQ	Question Answering	O	30
CommonsenseQA	Question Answering	O	30
MedQA	Question Answering	O	30
OpenBookQA	Question Answering	O	30
PIQA	Question Answering	O	30
Social IQA	Question Answering	O	30
TruthfulQA (MC1)	Question Answering	O	30
WinoGrande	Fill in a Blank	O	30
GSM8K	Mathematical Reasoning	X	30
MBPP	Code Generation	X	30
HumanEval	Code Generation	O	30
ANLI	Text Classification	O	30

Table 3. Test datasets sampled from Phi-3-medium-4k-instruct becnhmark datasets. Datasets with multiple choices have the instruction at the beginning. Each test dataset has 30 samples.

model (**'Pre + Inst'**) was obtained through instruction tuning after pretraining. The pretrained model (**'Pre'**) was trained solely on the pretraining dataset. Another model (**'Autoreg'**) was trained on an autoregressively generated dataset using the Phi-3 base model with ShareGPT inputs. For comparison, we also trained a model (**'Inst'**) that was initialized randomly and then fine-tuned exclusively on an instruction-augmented dataset (instruction tuning only). Lastly, the reproduced model (**'Rep'**) was trained on the fixed ShareGPT dataset.

A. *Experimental Setup*

We conducted experiments on various tasks, including question answering, code generation, mathematical reasoning, sentence completion, and text classification. For each task, 30 data points were randomly sampled from the benchmark datasets (see Table 3). Using the EAGLE-1 speculative decoding framework, we set a draft length of 5, top-k value of 15, and temperature to 0. The Phi-3-medium-4k-instruct (14B) model served as the base model, with all inferences run on a Jetson AGX Orin 64GB device. Inferences were performed with a batch size of 1, and the Hugging Face pipeline was the baseline for comparison.

B. *Results*

The **'Pre + Inst'** model demonstrated the best overall performance. While instruction tuning is generally expected to decrease draft accuracy for tasks lacking instructions (such as GSM8K and MBPP), the observed drop rate was not significant. In fact, for GSM8K (text2text generation) without instructions, the acceptance rate increased, while other tasks with instructions showed significant improvements (see Fig. 4 and Table 4). The only exceptions were the code generation tasks, MBPP and HumanEval, where **'Pre'** slightly outperformed **'Pre + Inst'**. Nevertheless, the **'Pre + Inst'** model maintained strong performance compared to the pretrained model, with only a

marginal drop, unlike the **'Inst'** model, which exhibited poor results on the code generation tasks. One notable observation is that the **'Inst'** model did not perform well even on tasks that included instructions. This suggests that instruction tuning is most effective when preceded by pretraining general knowledge without instructions. The draft models showed low acceptance rates on BoolQ and ANLI. This is because BoolQ's output is limited to just 2 tokens ("True" or "False"), and with a draft length of 5, the acceptance rate cannot exceed 20%. ANLI faces a similar challenge due to its short token generation length, which similarly restricts the acceptance rate.

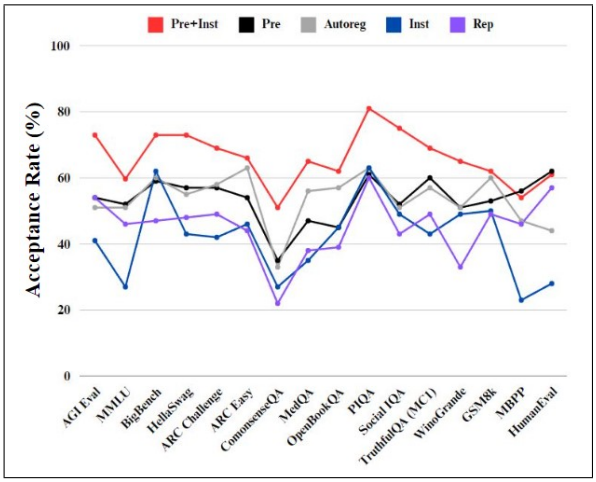

Fig. 4. Acceptance Rate of the draft models.

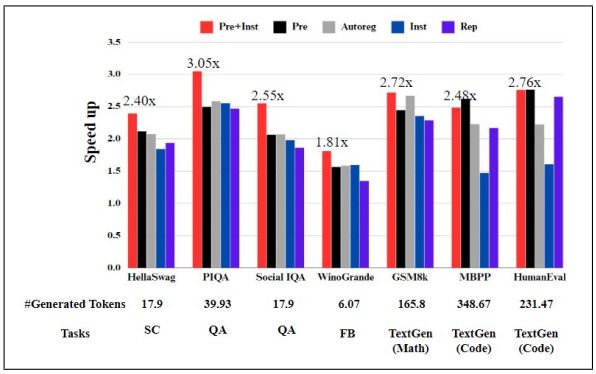

Fig. 5. SpeedUp of the draft models compared to the Hugging face Pipeline as the Baseline.

V. Conclusion

Our study highlights key findings for the EAGLE framework. First, the draft model should be trained on a dataset that closely resembles that of the base model. When the original dataset for the base model is unavailable, benchmark datasets can be an alternative. For instruction-tuned models like Phi-3, the best performance came from pretraining the draft model followed by instruction fine-tuning, leading to higher acceptance rates. Our proposed method broadens the options for base models within the EAGLE framework, allowing effective training of draft models even without access to the original dataset, especially for instruction-tuned models.

Test Datasets	Draft Models					Avg. Gen.
	Pre + Inst	**Pre**	**Autoreg**	**Inst**	**Rep**	
AGI Eval	72.94%	54.30%	50.87%	41.44%	53.65%	31.3
MMLU	59.68%	51.92%	51.10%	27.17%	46.34%	24.77
BigBench Hard	72.95%	58.82%	60.21%	61.94%	46.61%	12.7
HellaSwag	72.94%	56.71%	54.58%	42.92%	47.55%	17.9
ARC Challenge	68.86%	56.52%	58.37%	41.83%	48.69%	11.23
ARC Easy	65.84%	54.32%	62.54%	46.21%	44.40%	19.53
BoolQ	20.00%	20.00%	17.50%	20.00%	17.50%	2
CommonsenseQA	50.77%	35.00%	33.47%	26.79%	22.22%	4.37
MedQA	64.95%	47.33%	56.47%	34.55%	38.27%	13
OpenBookQA	62.45%	44.89%	56.99%	44.78%	39.23%	21.3
PIQA	81.10%	61.28%	63.19%	62.69%	59.87%	39.93
Social IQA	75.04%	52.45%	52.21%	49.29%	43.29%	17.9
TruthfulQA (MC1)	68.71%	60.42%	56.94%	43.36%	48.55%	12.57
WinoGrande	64.65%	51.11%	51.37%	48.68%	32.75%	6.07
GSM8K	62.25%	53.03%	59.42%	50.21%	49.04%	165.8
MBPP	53.51%	56.38%	46.58%	23.48%	45.52%	348.67
HumanEval	60.65%	61.69%	44.41%	27.62%	56.88%	231.47
ANLI	38.67%	22.38%	26.32%	38.67%	19.11%	2.93

Table 4. Average Acceptance Rate of Draft Models. Definitions: **(Pre + Inst)**: instruction tuning following pretraining; **(Pre)**: pretraining only; **(Autoreg)**: trained on autoregressively generated outputs from the base model using ShareGPT inputs; **(Inst)**: instruction tuning only; **(Rep)**: trained on a fixed dataset, ShareGPT. 'Avg. Gen.' refers to the average number of generated tokens per sample during decoding.

Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT)(2020-0-01297, Development of Ultra-Low Power Deep Learning Processor Technology using Advanced Data Reuse for Edge Applications)

References

[1] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv preprint arXiv:2302.01318*, 2023.

[2] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via speculative decoding. *arXiv preprint arXiv:2211.17192*, 2023.

[3] Yuhui Li, Fangyun Wei, Chao Zhang, and Hongyang Zhang. Eagle: Speculative sampling requires rethinking feature uncertainty. *arXiv preprint arXiv:2401.15077*, 2024.

[4] Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, et al. Instruction tuning for large language models: A survey. *arXiv preprint arXiv:2308.10792*, 2023.

SUMMARY OF THIS PAPER

A. *Related Works*

This paper expands on speculative decoding that aims to leverage a faster, lightweight draft model to predict multiple tokens in parallel, while minimizing the expensive computations required by the base model. The study focuses on the EAGLE (Extrapolation Algorithm for Greater Language-model Efficiency) framework, which improves drafting accuracy by incorporating both feature and token sequences during the drafting phase. A key factor in speculative decoding performance is the acceptance rate of a draft model. A higher acceptance rate reduces the number of tokens that need to be reprocessed by the base model, leading to fewer forward passes and overall lower computational overhead.

B. *Problem Setup (Challenges)*

The EAGLE framework requires a draft model to align with the base model, as the draft model must predict the next feature (or last hidden state) of the base model. As a result, training a draft model specifically for each base model is necessary. A major challenge arises when the original dataset used to train the base model is unavailable. This poses a problem because the draft model must closely match the base model's behavior to achieve high acceptance rates.

C. *Novelty*

This paper presents key findings on the EAGLE framework and introduces a novel training method that does not need access to the original training dataset of the base model. Contrary to the claims made in the original EAGLE paper, which suggested EAGLE's low sensitivity to training data, we demonstrate that when there is a significant mismatch between a training dataset and a base model, the draft model's performance is affected. For this reason, the choice of a base model in the EAGLE framework was previously limited by the unavailability of its training data. Our proposed method, which includes pretraining and instruction fine-tuning on open-source benchmark datasets, allows the draft model to align closely with the base model, expanding the range of base models that can be used within the EAGLE framework.

D. *Training Methods*

When the original dataset for a base model is unavailable, training on benchmark datasets where the base model performs well can be a viable alternative. The proposed training process involves two phases: pretraining and instruction fine-tuning. In the pretraining phase, the draft model was trained on benchmark datasets without augmentation to acquire general knowledge and improve its ability to generate drafts for the base model's unpredictable outputs. In the instruction fine-tuning phase, instruction-augmented datasets were used to help the draft model align more closely with the base model's characteristics, particularly because the base model used in this study was instruction-tuned.

E. *Experiments*

The experiments involved five different draft models trained on a variety of datasets. The draft models were tested on various tasks such as question answering, code generation, mathematical reasoning, and text classification. The model trained using the proposed pretraining and instruction fine-tuning method demonstrated the highest acceptance rates across diverse tasks. Notably, this approach improved acceptance rates even on unseen datasets, outperforming other approaches.