Optimizing Performance through Different Fine-Tuning Techniques in Small Language Models

Alan Yang
Student
Department of Computer Science
McGill University
kai.y.yang@mcgill.mail.ca

Michael Xu
Student
Department of Computer Science
McGill University
jiazheng.xu@mail.mcgill.ca

Abstract

This study investigates the enhancement of Small Language Models (SLMs) through advanced fine-tuning techniques, focusing particularly on Microsoft's Phi-2, a model noted for its efficiency in a compact size. Unlike prevailing research which predominantly targets Large Language Models (LLMs), this research shifts attention to SLMs, aiming to demonstrate their capability to perform tasks traditionally associated with their larger counterparts. This shift is significant as it challenges the notion that only LLMs can excel in complex tasks such as in-context learning and algorithmic reasoning, potentially democratizing access to powerful AI tools.

At the heart of our research is the application of instruction tuning and symbol tuning to SLMs. These methods have shown effectiveness in enhancing LLMs, and we hypothesize they will similarly improve SLMs' performance. Our goal is to present substantial advancements in SLM task execution, advocating for a new direction in natural language processing that prioritizes cost-effective and accessible AI technologies. We expect our findings to encourage further exploration into SLMs as practical alternatives for various applications to narrow the divide in AI accessibility and efficiency. The code, models and datasets are publicly available [1].

1 Introduction

1.1 Phi-2 Small Language Model

Small Language Models (SLMs) like Phi-2, developed by Microsoft, have emerged as efficient alternatives to their larger counterparts, combining quicker inference times and reduced computational demands. Despite their advantages, SLMs face challenges in handling complex tasks due to their constrained size, highlighting the need for innovative fine-tuning techniques. Inspired by Wei et al. (2022), who demonstrated how symbol and instruction tuning significantly enhance LLM's effectiveness. Our study aims to adapt these techniques for SLMs and seeks to bridge the performance gap, leveraging the compact nature of models like Phi-2 while addressing their limitations. We explore the potential of these tuning methods to generalize and elevate SLM performance in different NLP tasks.

1.2 Instruction Tuning

Instruction tuning technique forces language models to adeptly follow explicit instructions, enhancing their performance on specific tasks. For example, it enables a model to precisely "Summarize the article" by focusing on extracting key points, leveraging the model's learned knowledge for task execution. However, instruction tuning falls short in ambiguous situations or when instructions are vague, limiting the model's generalization capabilities. These constraints highlight the importance of symbol tuning, which, by replacing natural language labels with symbols, encourages models to infer and generalize from context rather than relying on explicit instructions. Integrating instruction tuning with symbol tuning aims to bridge these gaps, offering a pathway to models that are not only instruction-compliant but also adept at navigating complex, less-defined tasks.

1.3 Symbol Tuning

Symbol tuning revolutionizes model training by substituting natural language labels with arbitrary symbols, compelling models to derive understanding from the structure and context of data rather than predefined instructions. This method enhances a model's ability to generalize across tasks by focusing on the relational dynamics between inputs and outputs, rather than the semantic content of labels. Figure 1 shows a comparison between sym-

[1] Code is available at https://github.com/alan5543/phi-2-tuning

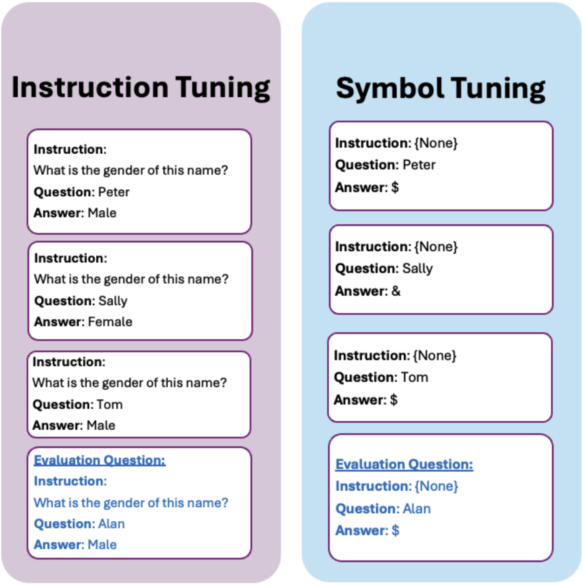

Figure 1: Comparison Between Instruction and Symbol Tuning

bol tuning and instruction tuning, the gender label can be represented by arbitrary symbols, such as Male is "$" and Female is "&". Given three exemplars in the prompt, the model can recognize the gender of "Alan" belonging to "$". The model learns to associate these symbols with their corresponding sentiments through the context in which they appear, not the inherent meaning of the gender words. This abstraction fosters deeper learning and adaptability, allowing models to better handle ambiguous or novel scenarios by focusing on pattern recognition rather than linguistic cues, thus broadening their applicability.

1.4 Our Workflow

Our workflow, visualized in Figure 2, commences with Phi-2 as the baseline model, setting the stage for a methodical enhancement through fine-tuning techniques. The journey begins by applying instruction tuning to Phi-2, transforming it into Flan-Phi-2, where it learns to navigate tasks with explicit instructions more adeptly. Building upon this foundation, we introduce symbol tuning to the now instruction-tuned Flan-Phi-2, advancing it to a version adept in symbol-based interpretation. It is designed to systematically bolster the baseline model's performance across key areas: in-context learning, algorithmic reasoning, and handling of flipped data scenarios. Through this strategic fine-tuning sequence, our study aims to explore and quantify the enhancements in the baseline model, potentially setting a new standard for the adaptability and efficacy of Small Language Models in complex NLP applications.

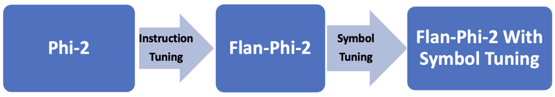

Figure 2: Progression of Phi-2 Through Fine-Tuning Techniques

2 Related work

2.1 Surprising performance of small language models

Benefiting from the high-quality training corpus generated by large language models such as GPT-3.5, small language models with fewer than 3 billion parameters can achieve performance comparable to models that have 25 times more parameters. For example, Eldan and Li (2023) demonstrated that the GPT-Neo model, with only 33 million parameters trained on 2 million synthetic stories, is capable of generating English sentences with a coherence level similar to that of GPT-4. By adding more synthetic training data and increasing the model's complexity, the Phi-1.5 model, with only 1.3 billion parameters created by Li et al. (2023), triples the accuracy on benchmarks such as GSM8K, HumanEval, and MBPP compared to larger models like Llama 2-7B. In our work, we explored two fine-tuning techniques to optimize Phi-2 from Javaheripi et al. (2023), which exhibits state-of-the-art performance among language models with less than 13 billion parameters.

2.2 Few-Shot learning ability

Current large language models, such as GPT-4.0, perform well even without examples given in the prompts, a technique known as zero-shot prompting. Wei et al. (2022) introduced the instruction tuning strategy, which can significantly improve the zero-shot learning capabilities of large language models like FLAN 137B. After being fine-tuned on over 60 NLP datasets transformed using instruction tuning templates, the performance of the Finetuned Language Net (FLAN) zero-shot model surpasses that of the GPT-3 175B few-shot model in several unseen tasks, including natural language inference, reading comprehension, and closed-book QA. Because we will use different symbols in the fine-tuning and evaluation stages to encourage the model to utilize its in-context learning ability, we must conduct few-shot prompting in the evaluation stage to teach the model the new symbols for different classes, even though it essentially remains a zero-shot learner. We assessed all three Phi-2 models on the same unseen NLP and algorithmic tasks to provide a fair comparison of their few-shot learning abilities.

2.3 Instruction tuning language models

Wei et al. (2022) mentioned that instruction tuning might hurt the generalization ability of small language models with fewer than 10 billion parameters on unseen tasks due to their limited capacity to learn all ~ 40 tuning tasks. On the other hand, Chung et al. (2022) demonstrated that instruction tuning can improve the performance of T5 and PaLM models with fewer than 3 billion parameters on unseen tasks in few-shot settings. We can conclude that the potential performance gain from instruction tuning is highly dependent on the model architectures and shot settings. Therefore, we used a subset of Chung et al. (2022)'s instruction tuning dataset to fine-tune the Phi-2 model, which has 2% of the number of parameters compared to the original FLAN, and evaluated its zero-shot learning ability on several unseen tasks. Given Phi-2's state-of-the-art performance on common reasoning and language understanding tasks as shown by Javaheripi et al. (2023), it should also represent the knowledge from instruction tuning datasets well.

2.4 Symbol tuning language models

Wei et al. (2023) discussed several drawbacks of instruction tuning. Firstly, models are encouraged to learn the semantic meaning of labels instead of their context. Secondly, instruction tuning degrades the untuned model's ability to make correct inferences following the flipped label in the prompts. The proposed symbol tuning technique from Wei et al. (2023) can mitigate these issues by forcing the model to conduct in-context learning only from the instructions and exemplars. Wei et al. (2023) also demonstrate that after being symbol-tuned on 5 datasets, adding more datasets does not lead to significant performance gains compared to the first five; thus, we only used a subset of the 22 tuning datasets from the original paper.

3 Modeling

3.1 Hypothesis and baseline

According to Wei et al. (2023), symbol tuning can enhance the in-context learning abilities of LLMs using significantly fewer data points compared to instruction tuning. However, Wei et al. (2022) stated that the average zero-shot accuracy on unseen tasks decreases after instruction tuning for models with fewer than 10B parameters, potentially because their capacity is insufficient to generalize the knowledge of tuning tasks to the unseen tasks. Wei et al. (2023) also mentioned the performance gain after symbol tuning in smaller language models, such as Flan-PaLM-8B, remains questionable.

The newly developed Phi-2 with 2.7B parameters, which has not undergone any instruction tuning, demonstrates superior reasoning and language understanding capabilities compared to the larger models discussed in the two aforementioned papers. Therefore, we hypothesize that both instruction tuning and symbol tuning strategies should help Phi-2 [2] follow the instructions more closely and enhance its in-context understanding despite its small size.

3.2 Flan-Phi-2

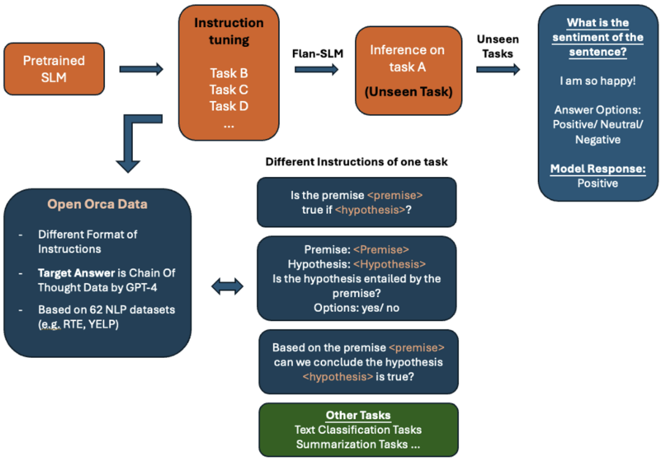
Pretrained SLM
Instruction tuning
Task B
Task C
Task D
...
Flan-SLM
Inference on task A
(Unseen Task)
Unseen Tasks
What is the sentiment of the sentence?
I am so happy!
Answer Options: Positive/ Neutral/ Negative
Model Response: Positive
Different Instructions of one task
Open Orca Data
- Different Format of Instructions
- Target Answer is Chain Of Thought Data by GPT-4
- Based on 62 NLP datasets (e.g. RTE, YELP)
Is the premise <premise> true if <hypothesis>?
Premise: <Premise>
Hypothesis: <Hypothesis>
Is the hypothesis entailed by the premise?
Options: yes/ no
Based on the premise <premise> can we conclude the hypothesis <hypothesis> is true?
Other Tasks
Text Classification Tasks
Summarization Tasks ...

Figure 3: Instruction tuning and evaluation

As shown in Figure 3, we conduct instruction tuning on Phi-2 using the Chain-of-thoughts OpenOrca Data from Lian et al. (2023) and evaluate it on tasks that were neither included in Javaheripi et al. (2023)'s training nor in our tuning stage, in order to reach an unbiased result. Through Chain-of-thoughts instruction tuning on a mixture of tasks,

[2] We used the publicly available Phi-2 from https://huggingface.co/microsoft/phi-2 in this project.

Flan-Phi-2 not only develops a richer contextual understanding of given prompts but also generalizes better on more challenging tasks through explicit reasoning compared to the baseline Phi-2.

3.3 Flan-Phi-2-Symbol

After developing Flan-Phi-2, we conducted symbol tuning on a small subset of Sileo (2023)’s data, using symbols as ground truth instead of traditional labels. To demonstrate the efficiency and effectiveness of symbol tuning, we tuned the model using only 8K rows of symbol tuning data, compared to the 14K rows of instruction tuning data for the Flan-Phi-2. During the tuning procedure, Flan-Phi-2-Symbol focused exclusively on pattern recognition between input-label pairs, as the arbitrary symbols carry no semantic meaning. This approach has improved the model’s in-context learning ability, as evidenced by the substantial performance boost across multiple benchmarks when evaluated on the same unseen tasks as Flan-Phi-2.

4 Dataset and evaluation

4.1 Fine-tuning datasets

4.1.1 Instruction tuning

We performed instruction tuning using randomly selected Zero-Shot Chain-of-Thoughts OpenOrca data from Lian et al. (2023), which is the Chat-GPT augmented version of the Flan Collection from Longpre et al. (2023). An example is as follows:

question: If "An older couple with joke glasses and cigars." does that mean that "A couple looking silly."? Options: - yes - it is not possible to tell - no Let’s be accurate as possible and think first.

response: Yes, "An older couple with joke glasses and cigars" would generally imply "A couple looking silly," as joke glasses and cigars are items typically used to create a humorous or lighthearted appearance.

Compared to the Flan Collection by Longpre et al. (2023), Mukherjee et al. (2023) keeps the answers and extends them with detailed step-by-step explanations using Chat-GPT. After tuning on the augmented version, much less tuning data is required for our model to reach the similar level of language understanding as the Wei et al. (2023)’s Flan model due to the richer signal in the OpenOrca data.

In the preprocessing process, we filtered out non-English examples because the Phi-2 model is pretrained only on an English corpus, and we built a tuning dataset with 140K examples. Each example consists of two columns: question and response. We concatenate these two for each example using the folloing prompt: *### Instruct: {example[’question’]}\n ### Output: {example[’response’]}* before sending to Phi-2. Additionally, any tuning example exceeding 512 tokens after the concatenation and tokenization is truncated to 512 to reduce VRAM usage.

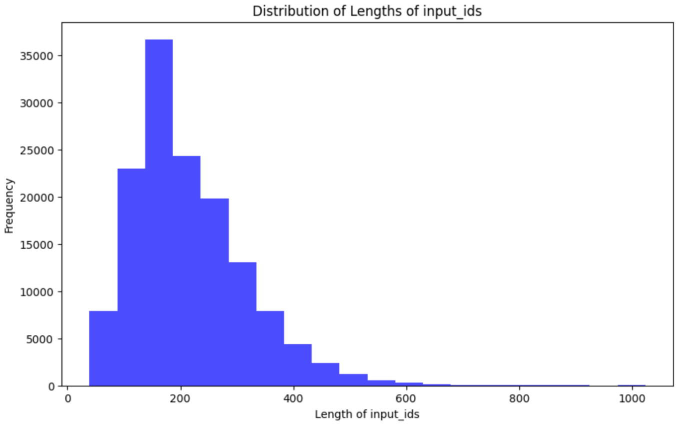

Figure 4: Length distribution of OpenOrca data after concatenation using prompt

For the sake of accelerating the tuning speed, we only include five examples in the validation set for the debugging purpose during the tuning process. A comprehensive evaluation will be performed manually after the tuning.

4.1.2 Symbol tuning

After instruction tuning, we selected a subset of 22 NLP classification datasets mentioned at Wei et al. (2023) and remapped the labels to symbols. An example is as following:

The original labels [“positive”, “negative”] have been remapped to [“1132”, “peter”].

Input: four-star movie Label: 1132
Input: can’t act Label: peter
Input: dumb , Label: peter
Input: not-so-funny Label:
Answer: peter

4.2 Evaluation Metrics

4.2.1 Natural language understanding

We selected four binary classification datasets from HuggingFace—TEH (Hate Speech Detection from Basile et al. (2019)), TEO (Offensive Language Detection from Zampieri et al. (2019)), SUBJ (Subjectivity from Antici et al. (2023)), and Climate (Topic Classification from Bingler et al. (2023))—which were not used during model tuning. For evaluation, we converted these datasets into an 8-shot format with equal positive and negative examples. We tested three formats: the first used correct labels with instructions, the second and third replaced labels with symbols, with only the second providing instructions. (see Appendix A). Symbols were assigned using an algorithm that randomly mapped labels to numbers, words, or characters.

4.2.2 Algorithmic reasoning

We utilized datasets from Big Bench from bench authors (2023), a benchmark designed to test language model reasoning. We focused on the Turing Test and various list functions (add elements, remove elements, input independent, modify list, miscellaneous). These datasets were converted into a 3-shot evaluation format, consisting of three examples followed by one question, as demonstrated below:

Drop the first and last elements.
Q: [4, 5, 0, 0] A: [5, 0]
Q: [3, 8, 3, 8, 3] A: [8, 3, 8]
Q: [5, 7, 7, 9, 8, 1, 4, 0, 6] A: [7, 7, 9, 8, 1, 4, 0]
Q: [2, 1, 1, 2, 2, 7, 2, 7] A:
Label: [1, 1, 2, 2, 7, 2]

4.2.3 Flipped labels task

For the Flipped Label evaluation, we used the same datasets as those in section 4.2.1 (TEH, TEO, SUBJ, and Climate), but with inverted labels: positive examples were mapped to negative, and vice versa. For instance, in the TEH tasks, the sentence 'I like you so much' was labeled as 'hate', while 'I want to complain about you' was labeled as 'Not Hate'. We formatted the data into 8-shot sets, comprising four positive and four negative examples, followed by a question for the model to answer.

5 Experiments

5.1 Common hyperparameters and settings

In both tuning procedures, we ran through the tuning data using one epoch, which was sufficient to demonstrate the effectiveness of the tuning strategies. Libraries such as datasets(v2.18.0), torch (v2.2.0) and transformers (v4.40.0.dev0) were used to load and tune the model. The Phi-2 model, loaded with 8-bit quantization prior to tuning, reduce VRAM usage significantly with almost no performance degradation, as stated by Belkada and Dettmers (2023).

Low-Rank Adaptation (LoRA) is applied to further ease GPU pressure because only a small scale of newly inserted weights are updated instead of all parameters. We set up the rank of the low-rank matrix used in the adapters at 64, alpha at 64, LoRA dropout rate at 0.05, and targeted all linear modules. Compared to the original settings from Hu et al. (2021), which used a rank of 64 and alpha of 16, we maintain the same rank setting to make sure the newly added linear layers are expressive and use a much higher alpha to force the model rely on the LoRA adapters more during inference. Thanks to the added LoRA layers, we reduced the trainable parameters from 2.8B to 94M and decreased the GPU's pressure by 96.7%.

Flash attention-2 from Dao (2023), with improved work partitioning on the GPU, is also utilized to increase tuning speed. We use an AdamW optimizer with 8-bit precision and a starting learning rate of 2.5×10^{-5}, which is adjusted from the beginning. Model uploading and logging occur every 25 steps through Weights & Biases. The code repository link can be found in the introduction section.

5.2 Flan-Phi-2

The hyperparameters include a training batch size of 2 and an evaluation batch size of 8. A single RTX 4090 can complete the tuning using 140K instruction data with a maximum token length of 512 in 10 hours.

5.3 Flan-Phi-2-Symbol

The hyperparameters include a training batch size of 4 and an evaluation batch size of 1. We reduced the evaluation batch size to free more VRAM for a larger training batch size, thereby increasing the overall tuning speed. A single RTX 4090 can complete the tuning using 8K symbol data with a maximum token length of 512 in 30 minutes.

6 Experimental Results

We conducted a human evaluation to verify our tuning techniques using data from three sections: 4.2.1 for evaluating in-context understanding, 4.2.3 for flipped label identification, and 4.2.2 for testing reasoning ability. Our baseline model was Phi-2, and evaluated two tuned versions: FLAN-Phi-2, which used instruction tuning, and FLAN-Phi-2-symbol, which combined instruction and symbol tuning. For a comprehensive comparison, we also tested a

	Phi-2 (Base)	FLAN Phi-2	FLAN Phi-2-Symbol
In Context Ability			
Classification with instructions and label	12%	63% (+51%)	52% (+40%)
Classification with instructions and symbol	15%	68% (+53%)	54% (+39%)
Classification with no-instructions and symbol	16%	39% (+23%)	49% (+33%)
Distinguish Flipped Label Ability			
Flipped Label Accuracy	19%	9% (-10%)	24% (+5%)
Algorithm Logic Ability			
Big Bench Accuracy	52.7%	54.2% (+1.5%)	46.7% (-6%)

Table 1: Performance Comparison of Phi-2 and the Fine-Tuned Versions

smaller language model, Mistral-7B-Instruct, and a larger language model, ChatGPT-3.5.

6.1 In Context Understanding Ability

Given an instruction and correct label, table 1 indicates that the base model Phi-2 has a 12% accuracy, but tuning significantly enhances performance, with FLAN-Phi-2 and FLAN-Phi-2-Symbol achieving improvements of +51% and +40%, respectively. Notably, Figure 5 demonstrates that in the TEH dataset for identifying hate speech, FLAN-Phi-2 surpasses both Mistral-7B and GPT-3.5, despite its smaller size of 2B compared to 7B and 175B. Instruction tuning markedly boosts the model's in-context understanding, with FLAN-Phi-2 generally outperforming FLAN-Phi-2-Symbol in both the TEH and Climate datasets when provided with precise instructions.

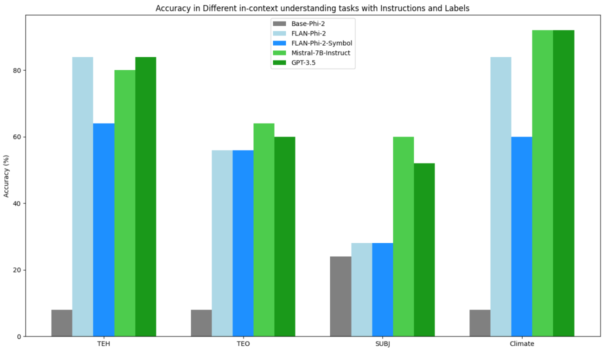

Figure 5: Given Instruction and Correct Label

Given an instruction and symbol, we replaced the classification label by an arbitrary symbol. Table 1 shows FLAN-Phi-2 achieved 68% accuracy and FLAN-Phi-2-Symbol 54%, both nearly quadrupling the base model's performance. Figure 6 indicates that FLAN models surpass Mistral-7B and GPT-3.5 in the TEO dataset and perform similarly in the TEH dataset. In comparisons across the TEO and Climate datasets, the models evolved from zero to approximately 70% accuracy. While the Symbol-Tuning model generally underperforms compared to the FLAN model, it significantly improves upon the base model and even exceeds FLAN in the SUBJ dataset. Both tuning methods substantially activate the base model's understanding.

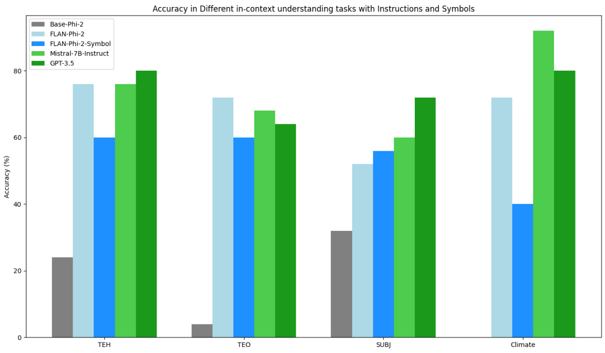

Figure 6: Given Instruction and Arbitrary Symbol

Without any instruction and using only symbol-based input-output pairs, where inputs are classification sentences and outputs replace labels with symbols, Table 1 reveals that FLAN-Phi-2-Symbol outperforms FLAN-Phi-2, achieving 49% accuracy compared to 39%, with the base model Phi-2 at only 16%. Figure 7 illustrates that the Symbol-Tuning models consistently surpass the FLAN models across various datasets and even exceed the larger Mistral-7B-Instruct, nearing GPT-3.5 performance in SUBJ datasets. We can see without any instructions or not accurate instructions, the FLAN model will easily fail in generalization, leading the FLAN-Phi-2-Symbol is better this time. This highlights Symbol Tuning's advantage in overcoming the limitations of instruction tuning, particularly its ability to enable the model to understand the contextual relationships in input-output pairs without relying solely on direct semantic retrieval.

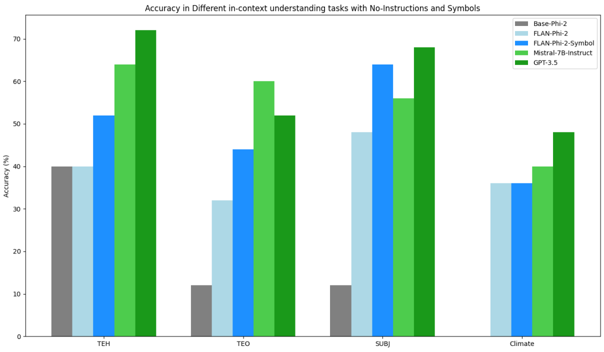

Figure 7: No Instruction and Arbitrary Symbol

6.2 Distinguishing Flipped Label

Table 1 reveals that FLAN-Phi-2 underperforms the base Phi-2 model by 10% when handling flipped labels, indicating a decrease in accuracy due to instruction tuning, which trains the model to strictly follow specific instructions or guidelines. This specific training causes the model to struggle with inverted labels. In contrast, FLAN-Phi-2-Symbol, which incorporates symbol tuning, significantly improves, achieving the highest accuracy at 24% for identifying flipped labels. Symbol tuning introduces an abstraction layer, enabling the model to focus less on the literal semantics of labels and more on recognizing patterns or relationships within the data. This makes FLAN-Phi-2-Symbol exceptionally effective, outperforming larger models like Mistral-7B and GPT-3.5, especially in the TEH dataset where its performance doubles that of these models and generally exceeds Mistral-7B in TEH, TEO, and SUBJ datasets, as shown in Figure 8.

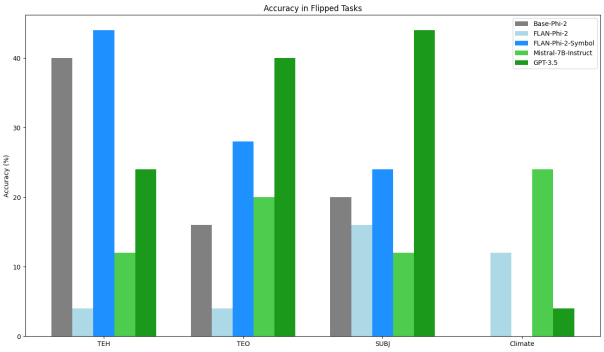

Figure 8: Accuracy of Identify Flipped Label

6.3 Algorithm Reasoning Ability

Using the Big Bench datasets for list function and Turing tests, FLAN-Phi-2 increased its accuracy by 1.5% over the base Phi-2, which holds an accuracy of 52.7%, while FLAN-Phi-2-Symbol saw a 6% drop. FLAN-Phi-2 consistently outperformed the base model in various logical tests, such as adding, modifying, and removing elements from lists. Moreover, FLAN-Phi-2 demonstrated superior performance to Mistral-7B across all test aspects, highlighting that instruction tuning enhances the model's capacity for reasoning and logic, particularly in list manipulation tasks. Despite having only 2B parameters, FLAN-Phi-2's post-tuning performance surpasses Mistral-7B-Instruct, showcasing its robust reasoning capabilities, though still trailing behind GPT-3.5.

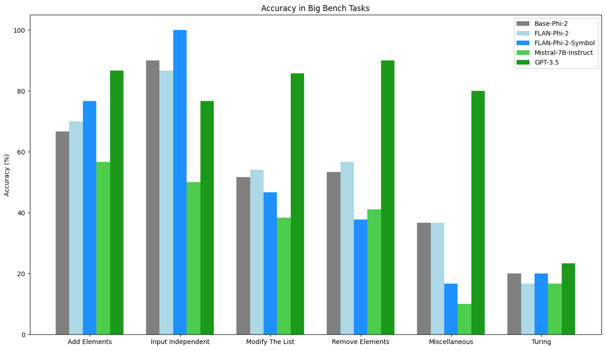

Figure 9: Accuracy of Big Bench tasks

Although FLAN-Phi-2-Symbol underperforms compared to the base Phi-2 and FLAN-Phi-2 in standard settings, it excels in input-independent tests, surpassing even GPT-3.5 in accuracy. These tests emphasize generalized reasoning and the ability to handle information without specific contextual cues. FLAN-Phi-2-Symbol's ability to interpret symbols rather than words enhances its adaptability and accuracy in scenarios requiring higher abstraction and generalization. Additionally, the symbol model slightly outperforms the base and FLAN models in Turing tests due to its enhanced capability to manage the ambiguity and complexity inherent in these evaluations, where intelligent behavior mimicking human interaction is crucial.

7 Discussions

7.1 FLAN enhance generalization of Phi-2

After instruction-tuning, FLAN-Phi-2 demonstrated enhanced capabilities in in-context understanding, algorithmic reasoning, and managing inverted labels, particularly when provided with specific instructions. This confirms our hypothesis that instruction tuning compels language models to meticulously follow explicit instructions, thereby boosting their task-specific performance. This technique proved effective in LLMs and our project showed its notable improvements in SLMs. By applying tuning to various tasks with tailored instructions, Phi-2 improved its generalization ability for handling unseen tasks. After tuning, the model is capable of generating both accurate and explainable outputs, such as answers with justifications:

"Input: Q: [9, 3, 6, 8] A: [8, 6, 3, 9] Q: [4, 0, 9] A: [9, 0, 4] Q: [9] A: [9] Q: [] A:

Output: A: []

Justification: Since there are no preceding inputs or outputs, the function can be applied to the final input list directly. In this case, the function simply returns the input list as it is."

7.2 Symbol Tuning when no instructions

Without adequate instructions, the FLAN model may struggle with generalization and task comprehension. However, we see the symbol model reached a higher accuracy in classification without instruction, inverted labelled tasks, and input-independent logical tests. There is a good testing result shown in Appendix B, given a simple input-output pair without giving any instructions, and we replaced the label with a random integers. The situation wants the model to find out the inner relationship between input and output, without giving it any clues of the tasks.

Symbol tuning enhances performance in scenarios lacking specific instructions or in input-independent contexts due to its abstraction from literal text to focus on data's structural and relational aspects. By substituting words or labels with symbols, the model learns to identify and process patterns less tied to the actual content, allowing it to function effectively even without detailed contextual guidance. Traditional language models heavily depend on semantic interpretation, but Symbol Tuning reduce this dependency, making the model less reliant on word-based knowledge and more on the underlying relationships or pattern recognition.

7.3 Comparison with Other Language Models

Despite their smaller 2B size, models using instruction tuning and symbol tuning perform comparably to the 7B-sized Mistral-7B-Instruct. These models excel in specific scenarios, such as handling input-independent tasks with the Symbol model and managing flipped label tasks in the TEH dataset, significantly outperforming GPT-3.5. This superior performance is due to symbol tuning's ability to abstract beyond literal text, enabling effective pattern recognition and processing without detailed instructions. This capability allows these models to handle complex scenarios requiring flexibility and abstraction, where larger models like GPT-3.5 may rely more on direct semantic processing.

7.4 Limited on the classification task

Although symbol tuning can help the model in no-instruction settings and restore the model's ability to handle flipped labels, the tuning data must come from classification tasks. In fact, most of these tasks are binary classification tasks because the length of few-shot data (we use four shots during symbol tuning) for multiclass classification tasks would exceed the context window of SLMs. Most publicly available NLP datasets are not even classification tasks, such as the OpenOrca Data from Lian et al. (2023) used during the instruction tuning stage. It's possible that replacing the words with richer signal to symbols in a variety of NLP tasks and force the model perform in-context learning with less clues. The development of symbol tuning strategies beyond binary classification tasks can be explored in future studies.

8 Conclusion

From our research, we have demonstrated that both instruction tuning and symbol tuning can be effective not only for larger language models with more than 10 billion parameters but also for smaller ones such as Phi-2 2.7B. Once the base non-instructed model possesses sufficient knowledge and capacity, the in-context learning ability unlocked by both tuning methods can be substantial, as shown in Figure 6. Beyond comparing tuning strategies, Flan-Phi-2-Symbol achieves performance comparable to Mistral-7B-Instruct, which has twice as many parameters and arguably has been tuned through far more steps. Future work could involve experimenting with additional tuning strategies for Phi-2 or validating the effectiveness of symbol tuning on much smaller language models such as tinystories-33M from Eldan and Li (2023). Alongside the surprising learning power demonstrated by SLMs over the past year, further performance boosts could potentially be achieved using existing approaches that traditionally were only effective in larger models.

9 Contributions

Alan Yang: Design Experiments, write sections of the report, and build model tuning data and evaluation data, perform evaluation.

Michael Xu: Design Experiments, write sections of the report, and conduct model tuning and perform evaluation.

A The Dataset Example for Evaluating In-Context Understanding

The following example illustrates the data structure used to evaluate the in-context understanding of the language model. The first evaluation prompt provides an instruction with a correct label, the second replaces the label with a symbol, and the third includes no instruction but uses a symbol. These prompts are designed primarily to test the model's context understanding.

A.1 Given Instruction and Correct Label

Our Input:

Question: Label the following tweet based on whether it contains hate speech.

@user I am serious and don't like you.

Answer: Hate

Question: Label the following tweet based on whether it contains hate speech.

@user EU keen to strike deal with Muammar Gaddafi on immigration | World news | The Guardian

Answer: Not Hate

Question: Label the following tweet based on whether it contains hate speech.

@user I think Booker is a more hysterical woman than Kamala Answer: Hate

Question: Label the following tweet based on whether it contains hate speech.

@user I have no comments on it.

Answer: Not Hate

Question: Label the following tweet based on whether it contains hate speech.

Let's buy the Uniqlo jacket on the weekend, it is at a discount.

Answer:

Expected Output:

Not Hate

A.1.1 Given Instruction but replace label by Arbitrary Symbol

*We mapped the "Hate" to "thgh", "Not Hate" to "kkie"

Our Input:

Question: Label the following tweet based on whether it contains hate speech.

@user I am serious and don't like you.

Answer: thgh

Question: Label the following tweet based on whether it contains hate speech.

@user EU keen to strike deal with Muammar Gaddafi on immigration | World news | The Guardian

Answer: kkie

Question: Label the following tweet based on whether it contains hate speech.

@user I think Booker is a more hysterical woman than Kamala

Answer: thgh

Question: Label the following tweet based on whether it contains hate speech.

@user I have no comments on it.

Answer: kkie

Question: Label the following tweet based on whether it contains hate speech.

Let's buy the Uniqlo jacket on the weekend, it is at a discount.

Answer:

Expected Output:

kkie

A.1.2 Given No Instruction and with Arbitrary Symbol

*We mapped the "Hate" to "thgh", "Not Hate" to "kkie"

Our Input:

Input: @user I am serious and don't like you.

Output: thgh

Input: @user EU keen to strike deal with Muammar Gaddafi on immigration | World news | The Guardian

Output: kkie

Input: @user I think Booker is a more hysterical woman than Kamala

Output: thgh

Input: @user I have no comments on it.

Output: kkie

Input: Let's buy the Uniqlo jacket on the weekend, it is at a discount.

Output:

Expected Output:

kkie

B Good Generation From Symbol Tuning

Below is one of the testing result to classify the subjectivity performed by FLAN-Phi-2-Symbol. This is a simple input-output pair, "Subjective" mapping to "47484" while "Objective" mapping to "70454"

——-

Question: Every resource to soften its institutional features has been used, including motion-picture shows, concerts, an extensive library, pool tables, newspapers, magazines.” It is really a better place to live than many of the private homes taxed to support it.

Answer: 47484

Question: Children of variants “don’t grow relative to their parent unless they have an advantage.”

Answer: 70454

Question: Two-person crews are protected in this agreement, a critical safety issue that the carriers have been trying to attack.

Answer: 70454

Question: The promotion of it for many is an avocation, for increasing numbers it is a profession, and for a very great number of more or less trained men and women it is employment and livelihood.

Answer: 70454

Question: We imagine today that everything is done by the aggregate without the will to act of any of the individuals composing the aggregate.

Answer: 47484

Question: The pandemic and politics are two primary factors.

Answer: 47484

Question: Here is the intent only to show how unlike and differently motivated forces, economic, social and political, are tending together not only to swallow up the national income in government but also to produce a result which some intend and some do not.

Answer: 47484

Question: However, Mokdad warned that while the Biden administration’s projections remain feasible, the currently dominant variant – omicron – still remains very contagious.

Answer: 70454

Question: Such competition is embarrassing and unscientific from the common point of view of government seeking revenue.

Answer:

——-

Output: The answer to the first question is 47484.

References

Francesco Antici, Andrea Galassi, Federico Ruggeri, Katerina Korre, Arianna Muti, Alessandra Bardi, Alice Fedotova, and Alberto Barrón-Cedeño. 2023. A corpus for sentence-level subjectivity detection on english news articles.

Valerio Basile, Cristina Bosco, Elisabetta Fersini, Debora Nozza, Viviana Patti, Francisco Manuel Rangel Pardo, Paolo Rosso, and Manuela Sanguinetti. 2019. SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 54–63, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Younes Belkada and Tim Dettmers. 2023. A gentle introduction to 8-bit matrix multiplication for transformers at scale using hugging face transformers, accelerate and bitsandbytes.

BIG bench authors. 2023. Beyond the imitation game: Quantifying and extrapolating the capabilities of language models. *Transactions on Machine Learning Research*.

Julia Bingler, Mathias Kraus, Markus Leippold, and Nicolas Webersinke. 2023. How cheap talk in climate disclosures relates to climate initiatives, corporate emissions, and reputation risk. Working paper, Available at SSRN 3998435.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Alex Castro-Ros, Marie Pellat, Kevin Robinson, Dasha Valter, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Tri Dao. 2023. Flashattention-2: Faster attention with better parallelism and work partitioning.

Ronen Eldan and Yuanzhi Li. 2023. Tinystories: How small can language models be and still speak coherent english?

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models.

Mojan Javaheripi, Sébastien Bubeck, Marah Abdin, Jyoti Aneja, Caio César Teodoro Mendes, Weizhu

Chen, Allie Del Giorno, Ronen Eldan, Sivakanth Gopi, Suriya Gunasekar, et al. 2023. Phi-2: The surprising power of small language models. *Microsoft Research Blog*.

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

Wing Lian, Bleys Goodson, Eugene Pentland, Austin Cook, Chanvichet Vong, and "Teknium". 2023. Openorca: An open dataset of gpt augmented flan reasoning traces. `https://https://huggingface.co/Open-Orca/OpenOrca`.

Shayne Longpre, Le Hou, Tu Vu, Albert Webson, Hyung Won Chung, Yi Tay, Denny Zhou, Quoc V Le, Barret Zoph, Jason Wei, et al. 2023. The flan collection: Designing data and methods for effective instruction tuning.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4.

Damien Sileo. 2023. tasksource: Structured dataset preprocessing annotations for frictionless extreme multi-task learning and evaluation. *arXiv preprint arXiv:2301.05948*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Jerry Wei, Le Hou, Andrew Lampinen, Xiangning Chen, Da Huang, Yi Tay, Xinyun Chen, Yifeng Lu, Denny Zhou, Tengyu Ma, and Quoc Le. 2023. Symbol tuning improves in-context learning in language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 968–979, Singapore. Association for Computational Linguistics.

Marcos Zampieri, Shervin Malmasi, Preslav Nakov, Sara Rosenthal, Noura Farra, and Ritesh Kumar. 2019. Semeval-2019 task 6: Identifying and categorizing offensive language in social media (offenseval). In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 75–86.