

NumEval - Numeral aware language understanding

1st Kisela Julia

*Department of Computer Science
University of Applied Sciences
Karlsruhe, Germany
kiju1012@h-ka.de*

2nd Sand Fabian

*Department of Computer Science
University of Applied Sciences
Karlsruhe, Germany
safa1017@h-ka.de*

3rd Kuan-hua He

*Department of Computer Science
University of Applied Sciences
Karlsruhe, Germany
heku1011@h-ka.de*

Abstract—We address the limitations of large language models (LLMs) in numeral comprehension and propose improvements. It highlights the importance of numerical reasoning in applications such as finance and legal contexts. Exploring state-of-the-art models like FlanT5 and Phi-3-mini, we demonstrate that Phi-3-mini outperforms FlanT5 in accuracy due to its scale and logical reasoning capabilities by achieving 87% accuracy in the Quantitative Prediction (QP) task. In the Quantitative Question Answering (QQA) task Phi-3-mini reached 80% accuracy. Chain-of-thought (CoT) prompting and few-shot examples enhanced the models' ability to handle complex numerical reasoning tasks even further, achieving higher accuracies.

Our results further suggest that augmenting datasets with step-by-step solutions generated using advanced models such as GPT-4o can improve the training and performance of numeral-aware tasks. The research highlights the need for advanced model architectures, fine-tuning techniques, and enhanced datasets to advance numeral-aware language understanding.

Index Terms—numerical reasoning, prompting

I. INTRODUCTION

Recent advances in natural language processing (NLP) have led to the development of large language models (LLMs), demonstrating remarkable capabilities in understanding and generating human language. However, despite these advancements, numeral awareness is limited. Understanding numerical values and their contextual implications is crucial for many applications. For instance, in financial analysis, distinguishing between a 5% rise in revenue and a 50% rise can lead to vastly different strategic decisions.

Motivation

Numerical reasoning and comprehension are increasingly recognised as critical components of natural language understanding, particularly in contexts where quantitative information plays a crucial role. As language models evolve, the ability to accurately interpret and manipulate numerical data is essential for applications in fields ranging from finance to healthcare, where precise numerical insights can impact decision-making and outcomes. By introducing NumEval, we explore the current capabilities of current state-of-the-art LLMs and contribute new insights and strategies for improving their numerical reasoning abilities.

State of the art

Numeracy is one of the most current research areas on large language models. It requires a solid numerical understanding

and proper and reliable logical reasoning. There are various approaches to investigating the limitations and capabilities of numerical reasoning in LLMs. For one, Mubashara et al. [1], which explores numerical reasoning in tabular data, and Mirzadeh et al. [2] present a new benchmark dataset for mathematical reasoning to counteract data contamination. Also, Geva et al. showed that numerical reasoning is injectible in data generation. [3].

This paper contributes to this research by showing that LLMs are more capable of numerical awareness with different evaluation methods, like chain-of-thought prompting. Second, numerical awareness can be improved by adding some logical arguments.

II. PROBLEM STATEMENT AND DATA

The previous SemEval challenges mostly dug into analysing words within texts. These semantic analyses try to find the correct context of words or phrases with multiple meanings within a document. As the SemEval challenges progressed, a discovery was made: comprehending numerical values can significantly enhance performance in specific tasks. Consequently, they introduced NumEval into the challenge, establishing a testbed for evaluating the performance of the current high-performing models in numerically aware language comprehension [4]. There are three NumEval tasks: Quantitative Understanding (English), Reading Comprehension of the Numerals in Text (Chinese) and Numeral-Aware Headline generation (English).

A. Problem of Numeral aware language understanding

To understand the consequences of numerical awareness in current models, consider, for example, the statement, "stealing 10\$" compared to "stealing 10000\$" in a legal context. Here, the magnitude of the numerical value is the decisive factor for the court judgement, and a misinterpretation of the model can lead to a false conviction. Another example would be where one anticipates a 30% rise in stock prices versus a 3% rise. This nuance plays a vital role in fine-grained sentiment analysis.

With an understanding of the consequences of numeral-aware language understanding, we aim to improve current models by tackling the NumEval task of Quantitative Understanding. This task has three subtasks: Quantitative Prediction (QP), Quantitative Natural Language Inference (QNLI) and

Quantitative Question Answering (QQA). Our work will propose solutions for the QP and QQA tasks.

The Quantitative 101 dataset is used as data for these tasks. It combines one generated dataset (CND) and three benchmark datasets: Numeracy-600K [5], EQUATE, [6] and NumGLUE Task 3 [7].

B. Quantitative Predictions

The QP problem approaches numeral awareness by training models to estimate magnitudes correctly. This certainly is a difficult task, as the model has first to determine the masked number and then asses the magnitude of it. Like in the example at the beginning, an accurate understanding of magnitudes is valuable for generating the correct prediction.

Dataset: The dataset for the QP task is part of the Quantitative 101 dataset. It divides the Numeracy-600k dataset [5] into training, development, and test splits. We use the development dataset as a validation dataset. There are 480000 training points, 60000 test points and 60000 validation points.

Figure 1 shows an example of this dataset. It consists of a headline or a comment with a number in it. The number is then masked with a [NUM] token. The goal is to predict the magnitude of the masked number.

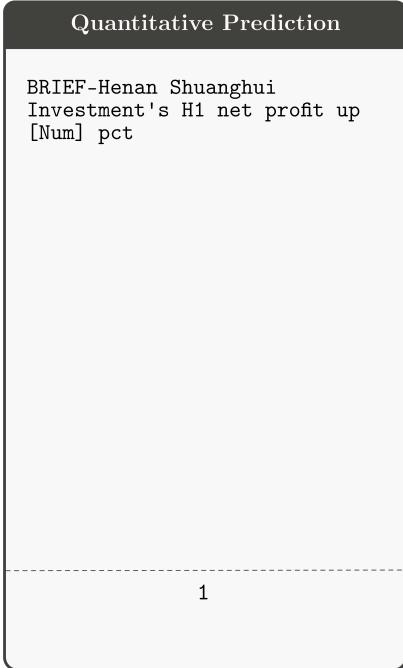


Fig. 1: Example of a datapoint of the QP task.

C. Quantitative Question Answering

QQA involves choosing the correct answer. The model receives a question and two options from which it must select

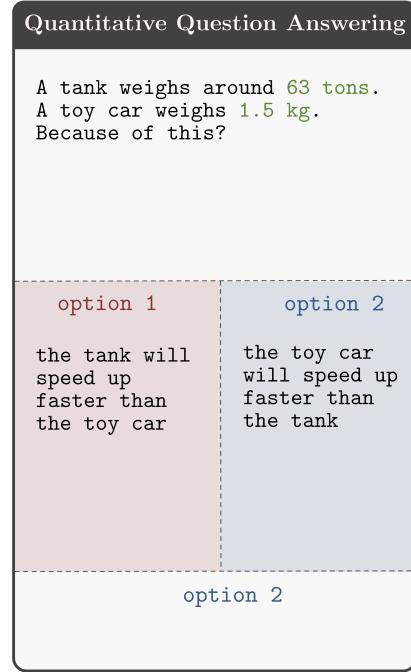


Fig. 2: Example of a datapoint of the QQA task.

the correct one. The question always involves two numbers, which must be compared to find the correct answer. To compare the numbers, the model also needs factual knowledge. The difficulty is that to solve the problem correctly, the models must combine language understanding, numerical awareness, and factual knowledge.

Dataset: The dataset comprises 564 training data points, 162 evaluation data points, and 82 test data points. Each data point consists of a question, two answer options, and the correct option.

Figure 2 shows an example of the QQA Dataset. At the top is the question, then the two options and at the bottom is the solution. To solve this example question, the model has to convert the units and compare the numbers so that the lighter object will speed up faster and will, therefore, look for the smaller number.

III. PROPOSED SOLUTION AND METHODOLOGY

The baseline established in this report originated in a paper submitted to the original SemEval 2024 challenge [8]. They worked on all three tasks of the NumEval challenge and delivered commendable achievements [8]. In their paper, the FlanT5 model demonstrated the highest performance in the original study, making the FlanT5-base [9] model a baseline for this research.

The decision to use the Phi-3-mini as our proposed model was based on the assessment that the task required strong numerical understanding, good general language comprehension,

factual knowledge, and sound logical reasoning, especially for QQA. These properties are already strong in most current large language models (LLMs) [2]. Due to the strong performance of the Phi-3-mini and computational limitations on our hardware, the Phi-3-mini is used as a proposed model for this study.

A. Quantitative Predictions

For QP, the model must predict a masked number's magnitude. The procedure involved generating a dataset with prompting, fine-tuning the baseline model, and comparing it to our proposed model. Each model is trained for five epochs on 1000 of the 480000 Datapoints. The models were then evaluated on the test data.

QP-Baseline model

For the reimplementation of the QP part using [8], the FlanT5-base implementation from Huggingface was used. FlanT5-base is an Encoder-Decoder Sequence-to-Sequence model developed by Google [9]. It has 250 Million parameters and can be finetuned for specific tasks. The preprocessing involved prompting with some few-shot examples to improve the model predictions. We use the same template with the exact examples from the original paper to finetune the baseline model. Due to limited hardware resources, the model could not be fine-tuned on the entire dataset. Instead, the training for the QP task was done only on 1000 random data points.

Proposed model

Phi-3-mini is a causal language model from Microsoft with a decoder-only architecture and 3.8 billion parameters [10]. A causal model is an autoregressive model that predicts the next token given only the previous tokens. These models are used for text generation, summarisation, or question-answering. In contrast, sequence-to-sequence models encode the input sentence, and the decoder generates the output. The models manage translation, summarisation, and question-answering.

The proposed Phi-3-mini was fine-tuned like the FlanT5-model on the 1000 Datapoints for 5 Epochs and with few-shot examples. For the implementation, the Huggingface interface of the Phi-3-mini served as the foundation.

B. Quantitative Question Answering

The QQA task consists of questions with two answer options, of which the model has to choose the correct option. A baseline and a second model were fine-tuned to solve the task on this dataset. Each model was trained for five epochs and then evaluated on their accuracy on the test data. The goal was to determine an already good model and then attempt to improve it further. For further improvement, we used few-shot examples, where the model is given exemplary questions and answers as part of the input, and chain-of-thought prompting, where the model is asked to solve a task step by step and deduce the answer logically. Both few-shot examples and chain-of-thought prompting have proven to be capable of improving the results of our proposed Phi-3-mini model.

Baseline model

Following the approach in the previous task, the FlanT5-base model is used as a baseline model. It was fine-tuned on the QQA Dataset, which required changes in the instruction template. The model was fine-tuned for five epochs on the QQA training dataset.

Proposed model

As was done in the preceding task, we chose Phi-3-mini for our proposed model. We first fine-tuned the Phi-3-mini, like the FlanT5, for five epochs on the QQA dataset with the same changed template.

Modified evaluation

As the Phi-3-mini performed better than the FlanT5, we continued to improve it. We need to examine the loss to motivate further research. The loss [Fig. 10] only decreases for the first epoch on the training data, while the validation loss does not decrease. This strongly suggested that further training would not improve generalisation but rather lead to overfitting or no change at all. Therefore, our approach was to improve the prompting and the data rather than training for more than five epochs. We made two changes.

First, we introduced a chain-of-thought prompting to enable the model to deduce the correct answer logically rather than guess it. Figure 3 shows an exemplary input with a chain-of-thought prompt. For the accuracy proposal, we then checked if the generated answer contained the correct output option and if the wrong choice was not in the answer. A simple string comparison was used for this.

As a second proposal, we included three exemplary questions with human-generated answers as a few-shot examples [Fig. 4]. This was done to suggest a general structure for solving the questions to the model.

gpt and CoT-Dataset

Since the chain of thought prompting with few-shot examples proved to be the most successful approach, with an accuracy of 80%, we focused on improving it. Unfortunately, the QQA dataset only provides the correct answer and no step-by-step solution. This meant that finetuning the model on the chain-of-thought task was not possible. Instead of adding step-by-step solutions by hand, which would have been a significant and time-consuming task, we created the step-by-step solutions using GPT-4o. GPT-4o is a strong state-of-the-art LLM from OpenAI [11], [2]. We used OpenAIs API for GPT-4o to generate step-by-step solutions for the QQA dataset, creating a new dataset that allows training with chain-of-thought reasoning. Fig. 5 shows an example of the dataset. Due to limited time and resources, the GPT-4o answers were not manually validated. Therefore, the new dataset contains inaccurate answers, and models trained on it can only be expected to achieve the same accuracy as GPT-4o. With this newly introduced dataset, the Phi-3-mini was again fine-tuned for the QQA task, this time with step-by-step solutions.

Using this approach also resulted in an evaluation of GPT-4o on the QQA dataset.

Quantitative Question Answering

<|system|> you are an AI assistant trained to answer Questions. You will decide whether " Option 1 " or " Option 2 " is correct. Think step by step<|end|><|user|> Marcus's son took the pool ball off the pool table to play with it around the house. The son noticed that the pool ball rolled a distance of 180 cm on the marble floor and 27 cm on the shag rug. The smoother surface is? **Option 1** : the marble floor or **Option 2** : the shag rug.. lets think step by step.<|end|>

Quantitative Question Answering

<|user|> Marcus's son took the pool ball off the pool table to play with it around the house. The son noticed that the pool ball rolled a distance of 180 cm on the marble floor and 27 cm on the shag rug. The smoother surface is? **Option 1** : the marble floor or **Option 2** : the shag rug.. lets think step by step.<|end|>

<|assistant|>To determine the smoother surface, we need to compare the distances the pool ball rolled on each surface.

Step 1: Identify the distances
Distance on marble floor: 180 cm
Distance on shag rug: 27 cm

Step 2: Compare the distances
The pool ball rolled a greater distance on the marble floor (180 cm) compared to the shag rug (27 cm).

Step 3: Determine the smoother surface
Since the pool ball rolled a greater distance on the marble floor, it experienced less friction and resistance. Therefore, the marble floor is the smoother surface.

the answer is: **Option 1**: the marble floor is the smoother surface<|end|>

Fig. 3: Example of our Chain-of-Thought evaluation.

Fig. 5: Example of the newly introduced got4 dataset.

Quantitative Question Answering

<|system|> you are an AI assistant trained to answer Questions. You will decide whether " Option 1 " or " Option 2 " is correct. Think step by step<|end|><|user|> Marcus's son took the pool ball off the pool table to play with it around the house. The son noticed that the pool ball rolled a distance of 180 cm on the marble floor and 27 cm on the shag rug. The smoother surface is? **Option 1** : the marble floor or **Option 2** : the shag rug.. lets think step by step.<|end|><|assistant|>To determine the smoother surface, we need to compare the distances the pool ball rolled on each surface.

Step 1: Identify the distances
Distance on marble floor: 180 cm
Distance on shag rug: 27 cm

Step 2: Compare the distances
The pool ball rolled a greater distance on the marble floor (180 cm) compared to the shag rug (27 cm).

Step 3: Determine the smoother surface
Since the pool ball rolled a greater distance on the marble floor, it experienced less friction and resistance. Therefore, the marble floor is the smoother surface.

the answer is: **Option 1**: the marble floor is the smoother surface<|end|>

<|user|>prompt2<|end|><|assistant|>solution<|end|>

....

<|user|>prompt4.. lets think step by step.<|end|>

Fig. 4: Example of our fewshot Chain-of-Thought evaluation.

IV. RESULTS AND COMPARISON

This section will compare the results of the FlanT5 baseline and the Phi-3-mini. For each task, we will look at the loss during the training and the accuracy of the test data achieved. We will then attempt to contextualise the result with the tasks and model.

A. Quantitative Prediction

For QP, the loss of the baseline model is shown in Fig. 6. While the loss on validation data is decreasing as expected, the training loss is curiously increasing before going down again.

The Phi-3-mini, on the other side, shows the expected drop in training loss, and the validation loss decreases as well[Fig. 7]. The low loss at the start for the validation data might be because the validation loss was measured after each training epoch. Therefore, the model was trained in one epoch before the first validation loss was computed. This leads to the false appearance of a lower validation loss than training loss. In Fig.8, the accuracy of the Phi-3-mini during training is plotted compared to the FlanT5 accuracy after training. This shows how much the Phi-3-mini model outperforms the FlanT5 model. This was expected to some degree, as the Phi-3-mini has 3.8 billion parameters and is, therefore, much bigger than the FlanT5 model with 250 million parameters. However, the large difference in the accuracy of 87% to 59% on the test data suggests that the task requires the stronger comprehension and logical reasoning abilities of the Phi-3-mini model. Even with

more extended training on the entire dataset, the FlanT5 model may not reach this accuracy.

B. Quantitative Question Answering

Fig. 9 and Fig. 10 show the training loss of the FlanT5 and Phi-3-mini model. Both plots show no decrease in the validation loss over the training time. This indicates that the training is not working properly, and the models are not generalising correctly. That is seconded by the low accuracy of the test data after the training.

The FlanT5 only achieves an accuracy of 50.9%, which is no better than guessing on a task with only two options. Even the stronger Phi-3-mini only achieved an accuracy of 67%, which is still not much better than random. This is also visible in Fig. 11, showing that the F1 scores increase slowly during the training and only around 0.5. As the validation loss is not decreasing, it is unlikely that further training would improve the models any further.

Figure 12 compares all accuracies of the different approaches on the QQA dataset. The most significant improvement comes from introducing chain-of-thought prompting to the Phi-3-mini. It improved by 40%, from 67% without to 80% with few-shot chain-of-thought prompting. This indicates that the questions were tricky and required multiple logical steps, which were difficult for the model to solve in one step. A closer examination of the questions also supports this. Many questions require converting numbers and demand an understanding of complex topics such as friction. The chain-of-thought prompting allows the model to solve the task in multiple steps and thereby logically deduce the correct answer, rather than having to know the correct answer after reading the sentence, which seems very difficult for this task. Learning such a general understanding from scratch would require massive datasets, which was impossible with the relatively small QQA dataset. Under these circumstances, it is not surprising that GPT-4o achieved the best result of all the tested models with an accuracy of 89%.

However, we showed that the Phi-3-mini Model's reasoning can be further improved for this task by training it with the step-by-step instructions generated by GTP-4o. Even though the answers of the GPT-4o model were not manually validated due to limited time and resources, they only had an accuracy of 89%.

V. CONCLUSION

This study explored the capabilities of numeral comprehension of Large Language Models. The study focused on two datasets, one for understanding the magnitude of numbers and the other for answering questions based on numerical comprehension. The performance of the baseline FlanT5 model was compared against the proposed Phi-3-mini model. The Phi-3-mini outperforms the FlanT5, showing its superior numeral comprehension and logical reasoning capabilities.

Key findings include the significant performance achieved through chain-of-thought prompting, improving the accuracy from 67% to 80% for the Phi-3 mini model. Furthermore, this

study shows the complexity of the questions and indicates the difficulty of finetuning a model on a task, requiring sound reasoning and general knowledge. As a solution, this work uses a workflow of generating step-by-step instructions using an even stronger model and learning the reasoning steps for the generated instructions.

Further work could be done to improve the newly generated dataset by filtering wrong answers. Interesting topics would also involve comparing more different LLMs on these tasks, finetuning the models on the full QP dataset and finetuning the models for more than five epochs. Numeral comprehension is still a big topic in AI, as even very strong LLMs like GPT-4o still struggle with the task of answering numerical questions correctly, as 89% accuracy is a good but far from perfect result.

CODE

The code for this paper can be found at:

<https://github.com/Fabian3112/NumEvalPhi3>

This repository contains a readme with instructions on replicating the results shown in this paper.

REFERENCES

- [1] M. Akhtar, A. Shankarapeta, V. Gupta, A. Patil, O. Cocarascu, and E. Simperl, "Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data," 2023. [Online]. Available: <https://arxiv.org/abs/2311.02216>
- [2] I. Mirzadeh, K. Alizadeh, H. Shahrokh, O. Tuzel, S. Bengio, and M. Farajtabar, "Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2410.05229>
- [3] M. Geva, A. Gupta, and J. Berant, "Injecting numerical reasoning skills into language models," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, D. Jurafsky, J. Chai, N. Schluter, and J. Tetreault, Eds. Online: Association for Computational Linguistics, Jul. 2020, pp. 946–958. [Online]. Available: <https://aclanthology.org/2020.acl-main.89/>
- [4] H.-H. H. Chung-Chi Chen, Jian-Toa Huang, "Numeral-aware language understanding and generation," in *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*, 2024.
- [5] e. a. Chen, Chung-Chi, "Numeracy-600k: Learning numeracy for detecting exaggerated information in market comments." in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. 2019.
- [6] e. a. Ravichander, Abhilasha, "Equate: A benchmark evaluation framework for quantitative reasoning in natural language inference." in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*. 2019.
- [7] e. a. Mishra, Swaroop, "Numglue: A suite of fundamental yet challenging mathematical reasoning tasks." in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. 2022.
- [8] X. Z. Kaiyun Chen, Jin Wang, "Instruction fine-tuning models for numerical understanding and generation." in *Proceedings of the 18th International Workshop on Semantic Evaluation (SemEval-2024)*.
- [9] H. W. Chung, L. Hou, S. Longpre, B. Zoph, Y. Tay, W. Fedus, Y. Li, X. Wang, M. Dehghani, S. Brahma, A. Webson, S. S. Gu, Z. Dai, M. Suzgun, X. Chen, A. Chowdhery, A. Castro-Ros, M. Pellat, K. Robinson, D. Valter, S. Narang, G. Mishra, A. Yu, V. Zhao, Y. Huang, A. Dai, H. Yu, S. Petrov, E. H. Chi, J. Dean, J. Devlin, A. Roberts, D. Zhou, Q. V. Le, and J. Wei, "Scaling instruction-finetuned language models," 2022. [Online]. Available: <https://arxiv.org/abs/2210.11416>
- [10] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao,

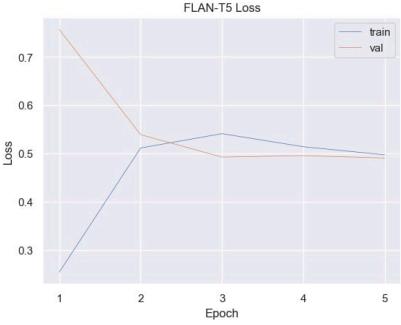


Fig. 6: Baseline loss plot of the QP task.

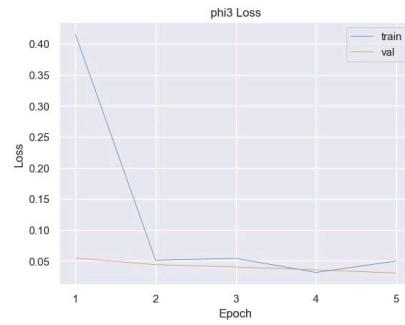


Fig. 7: Phi-3-mini loss plot of the QP task.

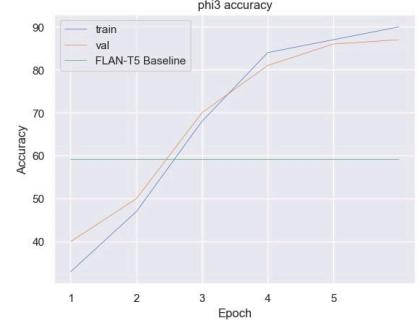


Fig. 8: Accucaryplot of the Phi-3-mini model versus the Baselinemodel.

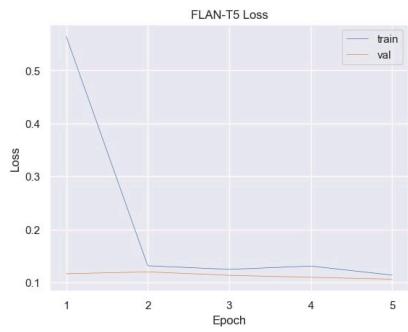


Fig. 9: Baseline loss plot of the QQA task.

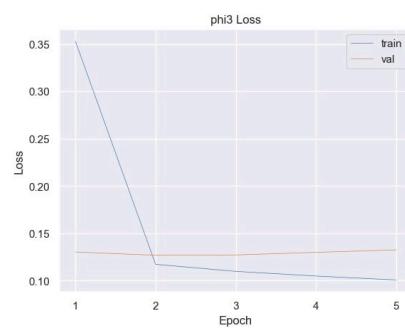


Fig. 10: Phi-3-mini loss plot of the QQA task.

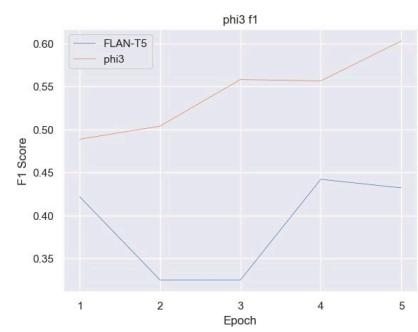


Fig. 11: F1-plot of the Phi-3-mini model versus the Baselinemodel.

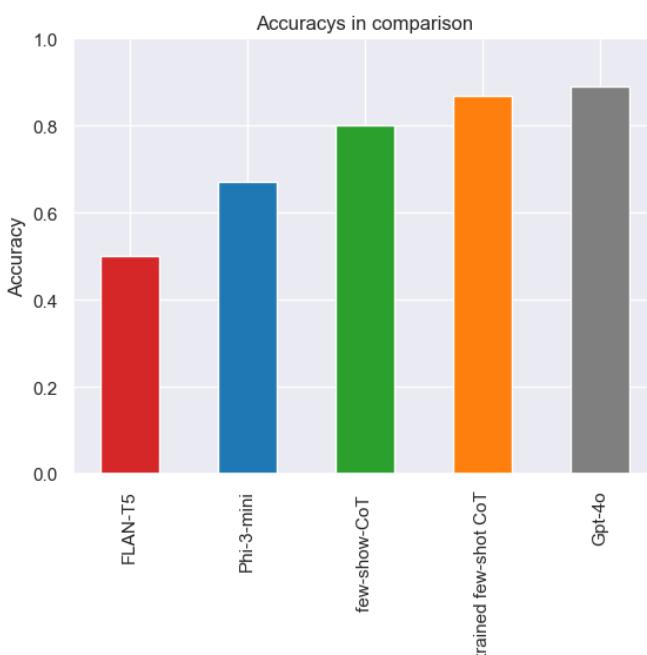


Fig. 12: Example of a figure caption.

R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatiakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saeid, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, and X. Zhou, “Phi-3 technical report: A highly capable language model locally on your phone,” 2024. [Online]. Available: <https://arxiv.org/abs/2404.14219>

[11] OpenAI, : et al., “Gpt-4o system card,” 2024. [Online]. Available: <https://arxiv.org/abs/2410.21276>