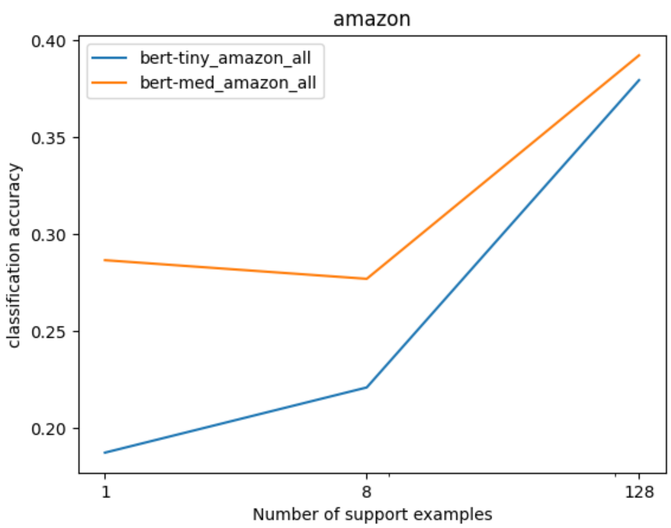
This handout includes space for every question that requires a written response. Please feel free to use it to handwrite your solutions (legibly, please). If you choose to typeset your solutions, the `README.md` for this assignment includes instructions to regenerate this handout with your typeset LaTeX solutions.

1.b (ii)

The accuracy is very poor $(< 40\%)$ for both of these models, with bert-med performing slightly better than bert-tiny. Couple of reasons

- The number of parameters in bert-tiny and bert-med is 4.4 million and 41.7 million, respectively - which is relatively very less
- Just a handful of examples $(1, 2, 128)$ is being used used to fine tune the entire set of parameters



1.c

- Number of parameters in Bert-mini = 11.2 million
- Space to store each parameter = 4 Byte floats

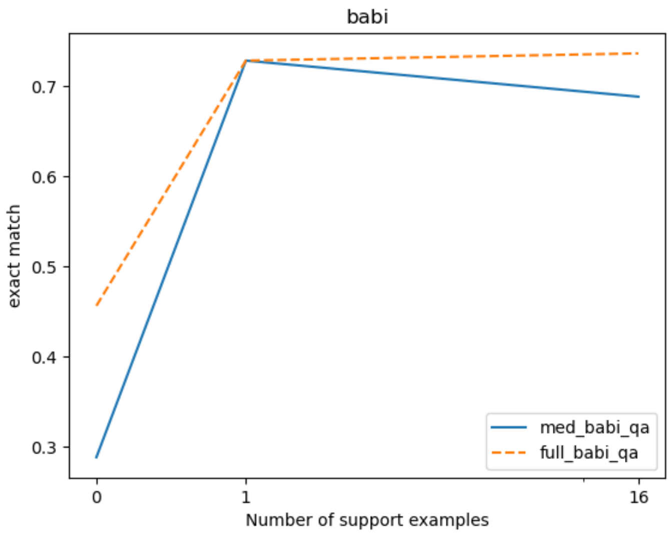
$\Rightarrow$ Total Disk space required to store the fine tuned model = 11.2 million $\times$ 4 bytes $= 44.8 \times 10^6$ bytes $= 44.8\ MB$

1.d

- Number of parameters in Google PaLM LLM = 540 billion $= 5.4 \times 10^9$

$\Rightarrow$ Disk space required to store a PaLM fine tuned model $= 5.4 \times 10^9 \times 4$ bytes $= 21.6\ GB$

2.b (ii)

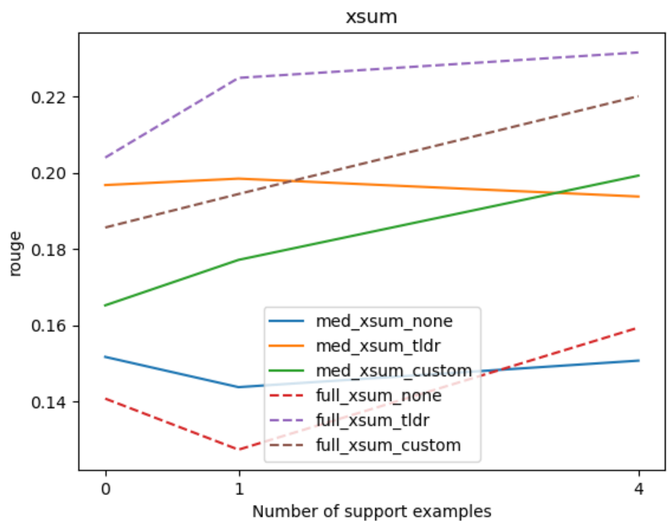


For the BABI dataset, 'qa' prompt is used which inserts ' In the ' in between every question and answer

Observations

- Larger model results in better performance (accuracy). Accuracy with full sized GPT2 model (with 1.5B parameters) at $k = 16$ shot $=$ 0.736, while the accuracy with med sized GPT2 model (355M parameters) at $k = 16$ is 0.688
- Accuracy generally increases with increasing k (number of examples used in the prompt) for both medium and full sized GPT2 model.

2.c (ii)



Observations

- **TL;DR:** prompt outperforms no formatting with a better rogue score. Since xsum dataset measures the performance of summaries, inserting a TL;DR: prompt would help the GPT2 model with a context on what needs to be done.
- Custom prompt: **Summarize in one sentence:** . Since the xsum dataset consists of labels which are summaries in a single sentence, this prompt was chosen. While the custom prompt yields a better rouge score than none prompt, it has a marginally lower score when compared to the TL;DR: prompt
- Generally, the performance improves with larger number of support examples, i.e., rouge score for zero-shot $<$ one-shot $<$ few-shot. However, there are couple of outliers in the above plot, e.g., a) TL;DR: prompt med model: 4-shot rogue score $<$ 1-shot rogue score b) none prompt (both med and full model): 1-shot rogue score $<$ zero-shot rogue score. These are probably due to smaller sized test set

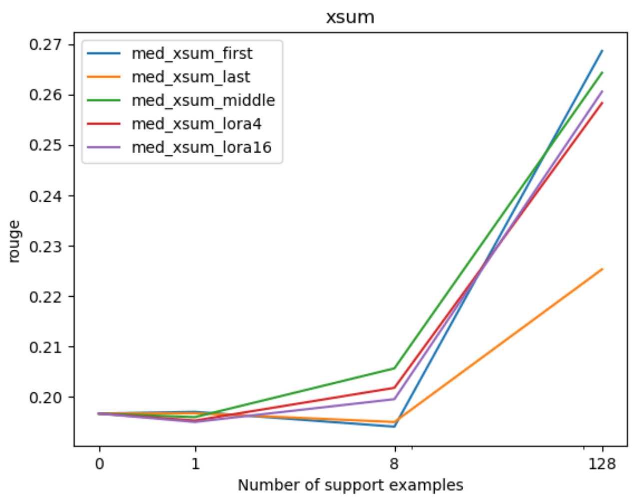
3.b

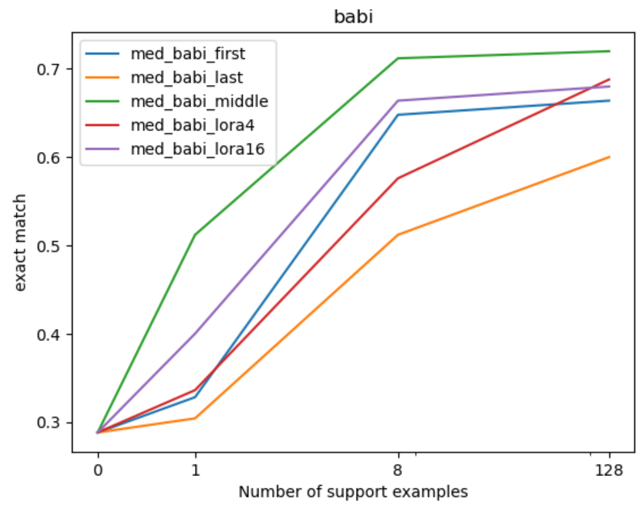
- Number of layers $= L$
- Number of parameters in the weight matrix W_0^l for each layer $= d_1 \times d_2$
- LoRA constrained fine-tuned parameter matrix $= AB^T$
- Size of matrix $A = d_1 \times p$
- Size of matrix $B = d_2 \times p$
- Total number of parameters that would be fined-tuned using LoRA $= d_1 \times p + d_2 \times p = p(d_1 + d_2)$

$\Rightarrow$ Ratio of the parameters fine-tuned by LoRA to the number of parameters in $W_l^0 = \frac{p(d_1+d_2)}{d_1 \times d_2}$

$\Rightarrow$ LoRA will provide the greatest savings in the newly-created parameters if $p << d_1, d_2$

3.d (ii)

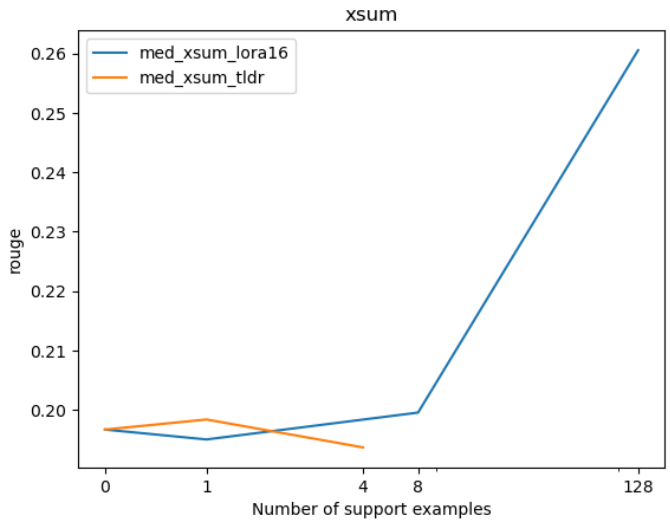




Observations

- Using a higher rank LoRA residual matrix results in better performance on both xsum and babi datasets (Comparing lora4 vs. lora16)
- LoRA outperforms the other first, last, middle fine-tune algorithms for larger number of support examples, though the number of parameters that are fine tuned as less

4.a



- In-context learning (tldr) seems like a better choice compared to fine-tuning (lora16) when there are very little support examples $(k = 0, 1)$.
- With higher number of support examples $(k = 4, 8, 128)$, fine-tuning (lora16) offers better performance over in-context learning (tldr)
- This highlights that in-context learning doesn't learn very well from support examples - as well as fine tuning. So, whenever we have more than a couple of support examples, in-context learning wouldn't benefit from these as much as fine-tuning the model would

4.b

Below table summarizes the exact match obtained for in context learning on BABI dataset for 5 random orderings of the prompt. As we can see, there is quite a deviation in these - varying from a low of 0.672 to a high of 0.728. The standard deviation for these $=$ 0.019

In-context learning would have a higher standard deviation than fine-tuning

Repeat Idx	Exact Match
1	0.688
2	0.728
3	0.712
4	0.696
5	0.672

4.c