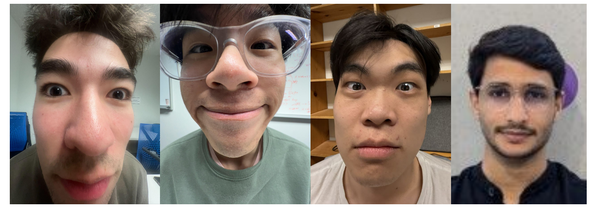
Prefix-Tuning: Optimizing Continuous Prompts for Generation

Mac Turner, Michael Ngo, Eric Hu, Neeraj Parihar
Cornell University



Problem & Motivation

You want GPT to do a specific task it's not directly trained to do. What are your options?

1. Fine-Tune for each task

For summarization

For data analysis

...

Too much compute 👎

2. Prompt Engineering

Summarize | this | text: | We're | gonna | win | so | much | you | may | even | ...

Analyze | this | data: | NBA | top scorers | LeBron | 42184 | | | Kareem | 38387 | | ...

...

Not expressive enough 👎

3. New! Prefix Tuning

Learned "Prefix" (passed in as K/V)

Input Tokens: NBA | top | scorers | LeBron | 42184 | | Kareem | ...

✅ Much less compute (0.1% of GPT-2 Params)
✅ More expressive than discrete prompt

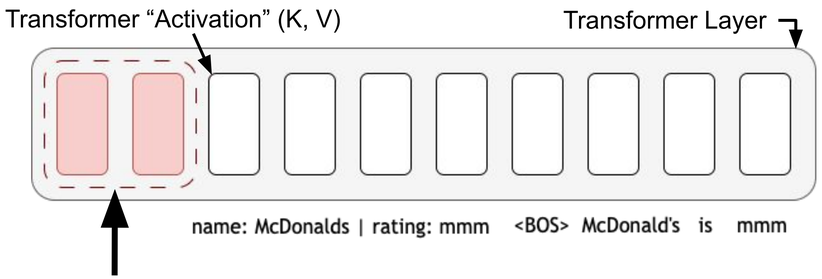
Contribution of Paper

1. Showed prefix tuning with 0.1% of model parameters of GPT-2, T5, BART is efficient and comparable to SOTA fine tuning.
2. Investigate prefix-length & position of prefix activations.

Our Goal

1. Reimplement prefix tuning and fine tuning for GPT-2.
2. Evaluate performance on the E2E table-to-text dataset.

What is the prefix?

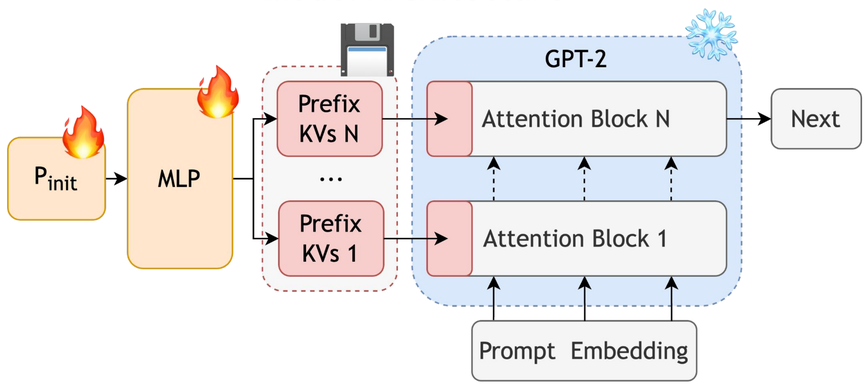


😐 **Tokens:** restricted to vocabulary, least expressive

🙂 **Word embeddings:** more expressive

😈 **Prefix activations (Learned keys + values)** : most expressive

Model Architecture



GPT-2 weights are frozen and prefix activations are learned via an MLP.
Only the output of the MLP is saved.

Dataset

"E2E Dataset": Table-to-text dataset about restaurants

name	type	food	price	rating	area	fam. friendly
The Mill	pub	Fast food	high	average	suburb	no

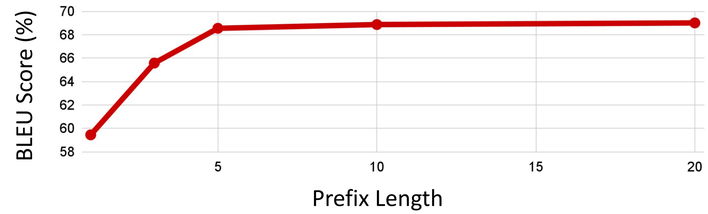
↳ "In the suburb is The Mill; a non-child friendly pub. Food is fast, price is high and average customer ratings."

Training

method	Learning rate	Epochs	Batch size	Prefix length	Training Time
Prefix(0.1%)	8e-05	5	10	5	10 m
Fine-tune	5e-05	5	10	-	17.5 m

Results

BLEU Scores with Different Prefix Lengths



Other Benchmarks

method	BLEU	NIST	METEOR	ROUGE-L	CIDEr
Prefix (0.1%)	68.5	**8.71**	44.3	70.4	2.34
Fine-tune	70.3	8.94	46.2	72.2	2.47
SOTA (2021)	68.6	8.70	45.3	70.8	2.37

Discussion

- Prefix-tuning did worse than fine-tuning - not due to generation, hyperparameters, dataset.
- Prefix-tuning is ~1.5x faster to train than fine-tuning

Future Work

1. Combination of prefix tuning with other parameter efficient fine-tuning methods
2. Special tokens to append task-specific prefix activations.

Conclusion

Prefix-tuning uses only 0.1% of memory to achieve SOTA. This can be used to efficiently fine-tune LLMs on many tasks with less memory without sacrificing performance.

References

Prefix-Tuning: Optimizing Continuous Prompts for Generation (Li & Liang, ACL-IJCNLP 2021)