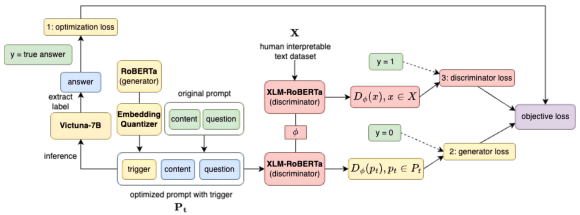
Enhancing Interpretability in Gradient-Based Prompt Optimization via GAN

Glenn Xu, Rolian Tan, Jennifer Guo
MSAII (Language Technologies Institute) & MSAIE (College of Engineering), Carnegie Mellon University

Motivation & Problem

- Gradient-based prompt optimization enhances LLMs via **trainable prefixes** optimized by gradient descent.
- Current optimized prompts are **unintelligible**, causing overfitting on limited data and reducing interpretability.
- We propose integrating **adversarial training** to address these challenges: use a GAN framework with a **generator** (optimized prompts) and a **discriminator** (evaluates interpretability) to iteratively balance **optimization** and **human interpretability**.

Method



The resulting discrete trigger tokens are represented as:

$$\mathbf{P_t}' = \text{Proj}_E(\mathbf{P_t}) := [\text{Proj}_E(\mathbf{t}_1), \ldots, \text{Proj}_E(\mathbf{t}_k)]$$

where Proj_E is the projection operation to the embedding space. During fine-tuning, we update the continuous trigger embeddings $\mathbf{P_t}$ using the gradient derived from the discrete vector $\mathbf{P_t}'$.

$$A = \theta(\mathcal{B}([\mathbf{P_t}, \mathbf{P}]), \mathbf{X})$$

where θ represents a downstream model (Vicuna-7B).

$$\mathcal{L}_{\text{optimization}} = -\sum_{n=1}^{N}\sum_{i=1}^{C} Y_i \log(f(A)_i)$$

where N is the num of samples, C is the num of classes, Y_i is the true label, f is the function to extract predicted label from model output.

$$\mathcal{L}_{\text{discriminator}} = \log(D_\phi(x))$$

$$\mathcal{L}_{\text{generator}} = \log(1 - D_\phi(p_t))$$

$$\mathcal{L} = \mathcal{L}_{\text{optimization}} + \mathcal{L}_{\text{generator}} + \mathcal{L}_{\text{discriminator}}$$

Results

Method	SST-2				Yelp P.			
	Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
Manual Prompts	0.66	0.67	0.64	0.65	0.72	0.69	0.80	0.74
PEZ	0.90	0.95	0.84	0.89	0.86	0.95	0.88	0.91
Ours	0.92	0.96	0.88	0.92	0.88	0.95	0.80	0.87

MR				AG's News			
Acc.	Prec.	Rec.	F1	Acc.	Prec.	Rec.	F1
0.56	0.78	0.67	0.72	0.48	0.59	0.73	0.65
0.78	0.89	0.82	0.85	0.74	0.90	0.73	0.81
0.86	0.91	0.80	0.85	0.80	0.86	0.72	0.78

Table 1: **Performance comparison across different methods on datasets.**

Method	Interpretability Score	ChatGPT-4o Accuracy (%)	Claude Accuracy (%)
Manual Prompts	0.95	0.73	0.61
PEZ	0.20	0.08	0.12
Ours	0.40	0.56	0.51

Table 2: **Generalization performance comparison across different LMs.**

Ablation Studies

1: Embedding quantizer w and w/ pre-warming

Method	Acc.	Prec.	Rec.	F1
w Pre-warming	0.87	0.92	0.80	0.86
w/ Pre-warming	0.70	0.78	0.75	0.76

Table 3: **Performance with Embedding Quantizer w and w/ Pre-warming.**
*** Pre-warming:** turn off embedding quantizer for first 3 epochs.

Without pre-warming:

```
resultkazykazykazytabkazy result
representation result Classification
resultkazyannotationÂerving
```

With pre-warming:

```
Tomatos consider Zone positive
encoding encoding Zone precise
Tomatos light Negative Tomatos Zone
encoding encoding Classification
```

Fig. 1: **Prefixes generated by Emb. Quantizer w and w/ pre-warming.**

Ablation Studies (cont'd)

2: Different methods of extracting labels

Method	Acc.	Prec.	Rec.	F1
Extract from Responses	0.87	0.92	0.80	0.86
Add Classification Layer	0.64	0.75	0.73	0.73

Table 4: **Average Performance with Different Methods of Extracting Classification Labels.**

Limitations & Conclusion

Limitations

- **Unstable GAN Training**: GAN-based methods are sensitive to initialization, requiring multiple runs to achieve optimal results, which increases computational costs. Future work could explore non-GAN approaches with better loss functions.
- **Vicuna Embedding Challenges**: Sparse token embeddings can get "trapped" after quantization, resisting updates despite pre-warming. Future research should improve adaptive quantization and pre-warming techniques.

Conclusion

We proposed a GAN-based framework for prompt optimization that enhances interpretability while maintaining strong performance. It outperformed baselines across datasets and showed strong generalization to different LLMs, including closed LLMs like ChatGPT-4o and Claude.

References

[1] Wen, Y. et al. Adv. Neural Inf. Process. Syst. 2024, 36.
[2] Li, X. L.; Liang, P. arXiv 2021, 2101.00190.
[3] Shin, T. et al. arXiv 2020, 2010.15980.
[4] Wallace, E. et al. arXiv 2019, 1908.07125.
[5] Du, Y.; Sun, W.; Snoek, C. G. M. arXiv 2024, 2410.15397.