You Only Prompt Once

Research Question

1. Can we use prompt learning to deal with toxic content?
2. Does prompt learning have advantages in performance and efficiency?

Prompt Engineering (https://arxiv.org/abs/2107.13586)

Pretrain, Finetune → Pretrain, Prompt, Predict

1. Prompt Addition
2. Answer Search
3. Answer Mapping

$$P(\boldsymbol{y}|\boldsymbol{x};\theta)$$

Prompt Addition

1. Apply a *template*, which is a textual string that has two slots: an *input slot* [X] for input $\boldsymbol{x}$ and an *answer slot* [Z] for an intermediate generated *answer* text $\boldsymbol{z}$ that will later be mapped into $\boldsymbol{y}$.
2. Fill slot [X] with the input text $\boldsymbol{x}$.

Can be viewed as modifying the input x to a prompt x'

$$\boldsymbol{x}' = f_{\text{prompt}}(\boldsymbol{x})$$

Answer Selection

$$\hat{\boldsymbol{z}} = \operatorname*{search}_{\boldsymbol{z} \in \mathcal{Z}} P(f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}); \theta).$$

Answer Mapping

Since we are looking for $P(\boldsymbol{y}|\boldsymbol{x};\theta)$

We are mapping the highest-scoring answer $\hat{\boldsymbol{z}}$ to the highest scoring output $\hat{\boldsymbol{y}}$.

Example

Name	**Notation**	**Example**	**Description**
Input	$\boldsymbol{x}$	I love this movie.	One or multiple texts
Output	$\boldsymbol{y}$	++ (very positive)	Output label or text
Prompting Function	$f_{\text{prompt}}(\boldsymbol{x})$	`[X]` Overall, it was a `[Z]` movie.	A function that converts the input into a specific form by inserting the input $\boldsymbol{x}$ and adding a slot `[Z]` where answer $\boldsymbol{z}$ may be filled later.
Prompt	$\boldsymbol{x}'$	I love this movie. Overall, it was a `[Z]` movie.	A text where `[X]` is instantiated by input $\boldsymbol{x}$ but answer slot `[Z]` is not.
Filled Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z})$	I love this movie. Overall, it was a bad movie.	A prompt where slot `[Z]` is filled with any answer.
Answered Prompt	$f_{\text{fill}}(\boldsymbol{x}', \boldsymbol{z}^*)$	I love this movie. Overall, it was a good movie.	A prompt where slot `[Z]` is filled with a true answer.
Answer	$\boldsymbol{z}$	"good", "fantastic", "boring"	A token, phrase, or sentence that fills `[Z]`

Prompt Learning Methods

1. Prompt Tuning: freeze the entire pre-trained model and only allow an additional k tunable tokens per downstream task to be prepended to the input text.

   For one task, one continuous prompt

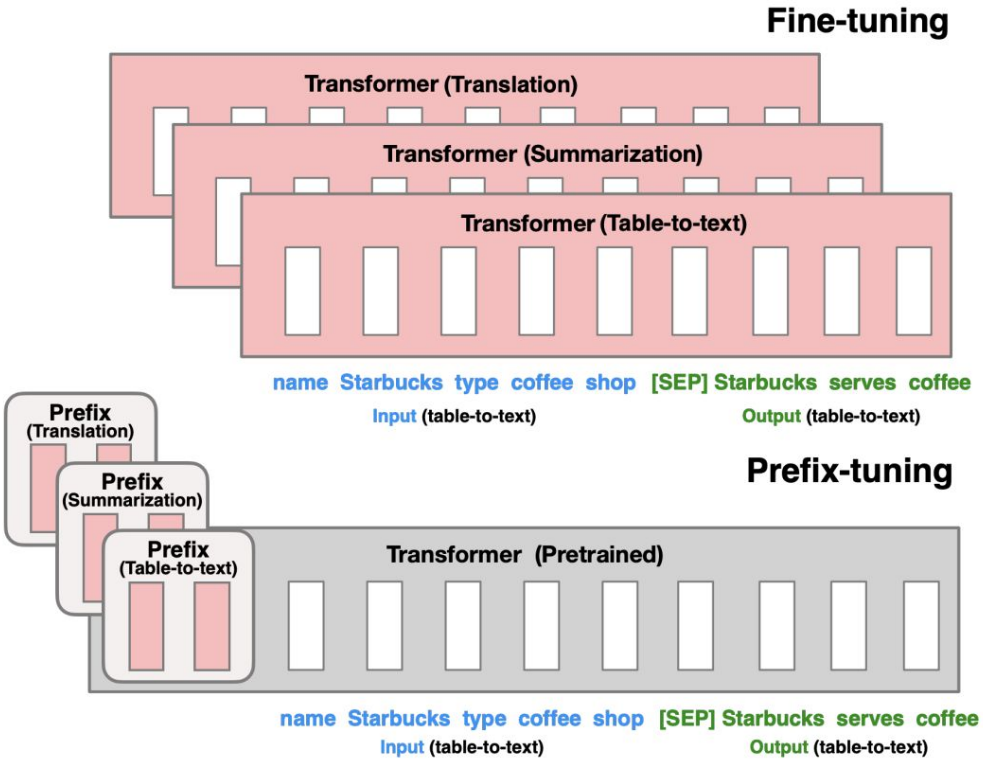
2. Prefix Tuning: For each layer of a Transformer, there is a prefix to tune.

   For one task, n prefixes, n = # layers of a Transformer.

   If all prefixes have a size of m, we need to learn m * n keys and m * n values

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Prompt Learning Methods



Toxicity Related Tasks

1. Toxicity Classification (binary)
2. Toxic Span Detection
3. Detoxification

Toxicity Classification	**Answer**
your reading comprehension is more fucked up than a football bat.	Toxic
Toxic Span Detection	**Answer**
keep hiring imbeciles like this jerk and you will end up with a no firearms for rent-a-cops bill next session.	keep hiring imbeciles like this jerk and you will end up with a no firearms for rent-a-cops bill next session.
Detoxification	**Answer**
what a chicken crap excuse for a reason.	what a bad excuse for a reason.

Formats of Predictions

1. Toxicity Classification: labels, benign vs toxic
2. Toxic Span Detection: texts without toxic spans, then toxic span = input text - output text
3. Detoxification: rephrased, non-toxic text

Choice of Prompt Learning Methods

1. Toxicity Classification: Prompt Learning
2. Toxic Span Detection: Prefix Tuning
3. Detoxification: Prefix Tuning

Evaluation Metrics

1. In-distribution (ID) performance
2. Out-of-distribution (OOD) performance
3. Robustness
4. Efficiency

Toxicity Classification

1. Compared to discrete prompt engineering, much better F1
2. Compared to fine-tuning, better F1
3. Can transfer to different datasets
4. Fewer training samples and training steps can have descent F1
5. Robust to adversarial perturbation

Table 3: F_1-score of Task 1. The best results of each dataset are highlighted in bold.

Dataset	Baselines			Prompt Tuning				
	Perspective	ToxicBERT	UnRoBERTa	GPT2-M	GPT2-L	T5-S	T5-B	T5-L
HateXplain	0.703	0.657	0.648	0.016	**0.731**	0.716	**0.731**	0.637
USElectionHate20	0.506	0.488	0.425	0.709	0.741	0.673	**0.833**	0.660
HateCheck	0.784	0.670	0.671	0.758	0.892	0.860	0.841	**0.946**
SBIC.v2	0.669	0.581	0.581	0.721	**0.854**	0.820	0.844	0.841
MHS	**0.790**	0.768	0.775	0.711	0.758	0.762	0.775	0.776
Avg.	0.690	0.633	0.620	0.583	0.795	0.766	**0.805**	0.772

Toxic Span Detection

Metric:

$$P^t(S_g^i, S_p^i) = \frac{|S_g^i \cap S_p^i|}{|S_p^i|}$$

$$R^t(S_g^i, S_p^i) = \frac{|S_g^i \cap S_p^i|}{|S_g^i|}$$

$$F_1^t(S_g^i, S_p^i) = \frac{2 \cdot P^t(S_g^i, S_p^i) \cdot R^t(S_g^i, S_p^i)}{P^t(S_g^i, S_p^i) + R^t(S_g^i, S_p^i)}$$

Toxic Span Detection

1. Comparable or even better F1 compared to fine-tuning
2. One training epoch can have descent F1
3. Not very robust to adversarial perturbation

Table 8: Performance of Task 2 (Toxic Span Detection).

Method	F_1	Time Cost (Second)
BiLSTM	0.566	94
BERT	0.629	1,828
SPAN-BERT	0.640	3,334
PT (T5-S)	0.571	175
PT (T5-B)	0.615	363
PT (T5-L)	0.643	838

Detoxification

Metric:

1. the average toxicity score change and the percentage of texts that has high toxicity score, returned by Perspective API
2. Fluency and semantic preservation

Results:

1. A little bit lower toxicity drop bug higher text quality
2. It is easier to generalize from bigger datasets to smaller ones
3. Robust to adversarial perturbation

Detoxification

Table 10: Performance of Task 3. The arrow denotes which direction is for better results.

Dataset	Method	$T_{avg} \downarrow$	$T_{0.7} \downarrow$	$T_{0.9} \downarrow$	BLEU $\uparrow$	SIM (W) $\uparrow$	SIM (F) $\uparrow$	TokenPPL $\downarrow$
Parallel	None	0.755	0.676	0.135	1.000	1.000	1.000	227.834
	GroundTruth	0.178	0.009	0.000	0.491	0.757	0.669	550.725
	BART	0.754	0.676	0.135	0.999	0.999	0.998	227.904
	DetoxBART	0.242	0.036	0.000	0.708	0.879	0.843	236.654
	PT (T5-S)	0.573	0.463	0.077	0.835	0.927	0.939	326.696
	PT (T5-B)	0.408	0.256	0.032	0.770	0.898	0.909	301.597
	PT (T5-L)	0.396	0.329	0.031	0.754	0.881	0.889	284.861
ParaDetox	None	0.775	0.778	0.134	1.000	1.000	1.000	330.829
	GroundTruth	0.166	0.000	0.000	0.633	0.828	0.778	393.800
	BART	0.774	0.777	0.133	0.999	0.999	0.998	331.250
	DetoxBART	0.180	0.013	0.000	0.688	0.862	0.832	438.242
	PT (T5-S)	0.253	0.081	0.007	0.760	0.910	0.905	593.442
	PT (T5-B)	0.224	0.051	0.005	0.754	0.920	0.897	499.851
	PT (T5-L)	0.213	0.037	0.003	0.743	0.916	0.886	404.565

Personal Opinions (Drawback of this paper)

1. No validation set.
2. All datasets are balanced (benign:toxic = 1:1), which is way not true in the real world. Sounds like cheating to me.