

Prompt-based Learning and Robustness Evaluation

Shuyue Jia

Ph.D. Student

Boston University

September 25th, 2023

Dependable Computing Laboratory,
Department of Electrical and Computer Engineering,
Boston University



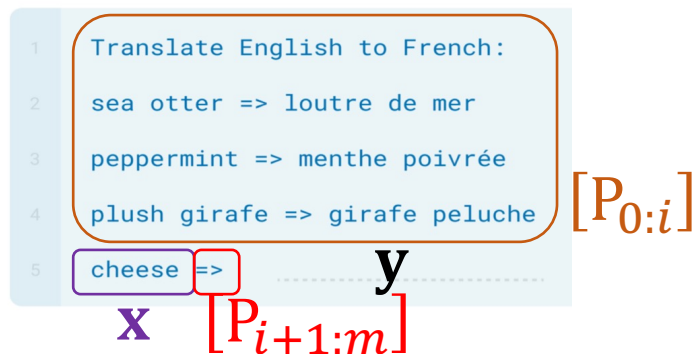
Outline

- Prompt Category
- **Continuous** Prompt-based Learning
- **Discrete** Prompt-based Learning
- Prompt **Perturbation and Robustness**
- Prompt **Robustness Evaluation**

Part 1 – Prompt Category

Few-shot

In addition to the task description, the model sees a few examples of the task. No gradient updates are performed.

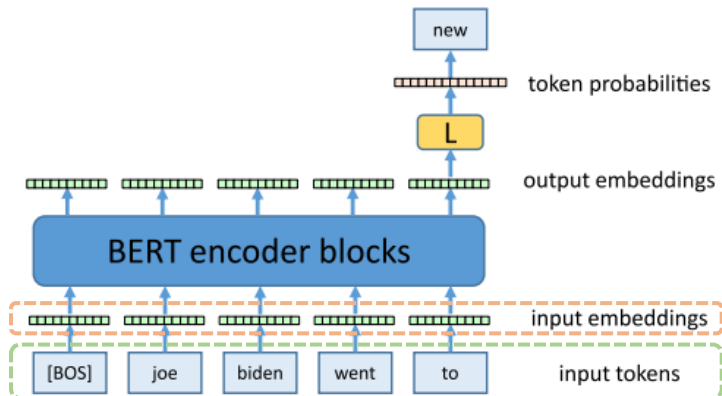
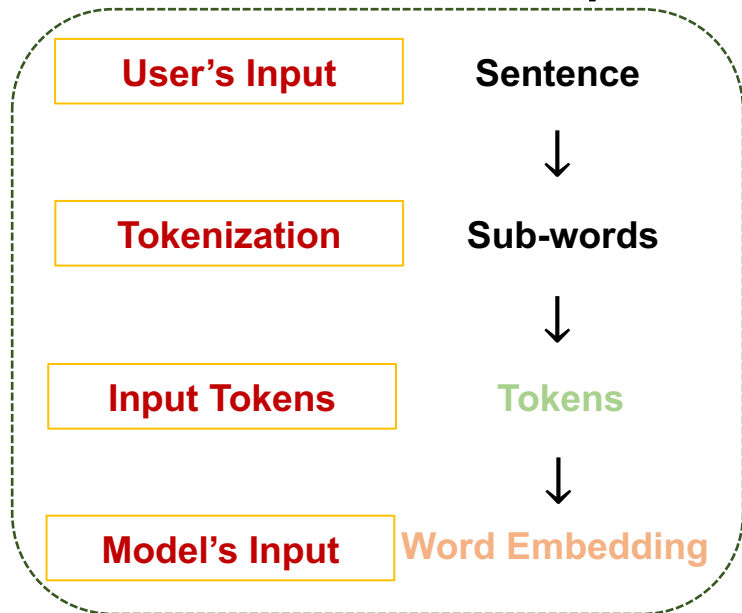


Template

$$T = \{ [P_{0:i}], \mathbf{x}, [P_{i+1:m}], \mathbf{y} \}$$

- **(Few-shot) In-Context Learning Imitation**
- **No gradient update** or **fine-tuning**
- Literally just take a pre-trained model and **give it the following prefix**

Part 1 – Prompt Category

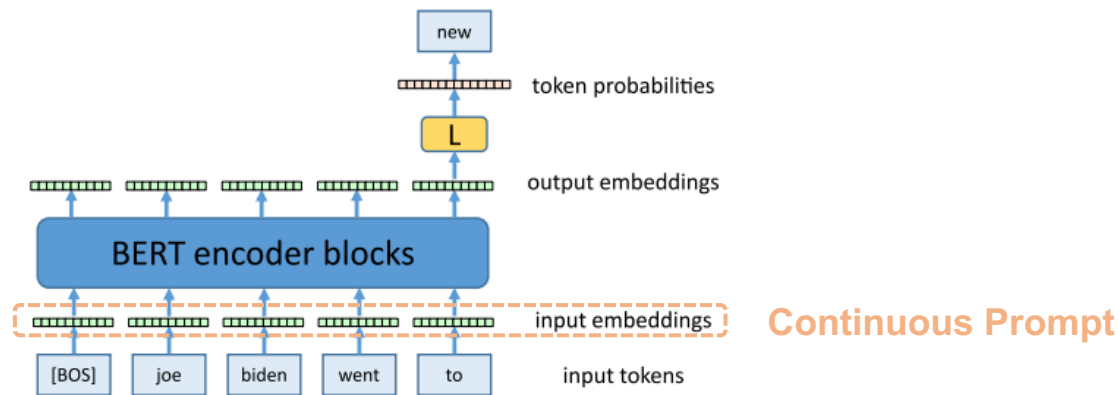


A Framework of Autoregressive LM

Continuous Prompt
Discrete Prompt

- **Discrete Prompt (Hard Prompt): Words**, learn lexical sequences
- **Continuous Prompt (Soft Prompt): Vectors**, learn embeddings (**Soften Version**)

Part 1 – Prompt Category – **Continuous Prompt**

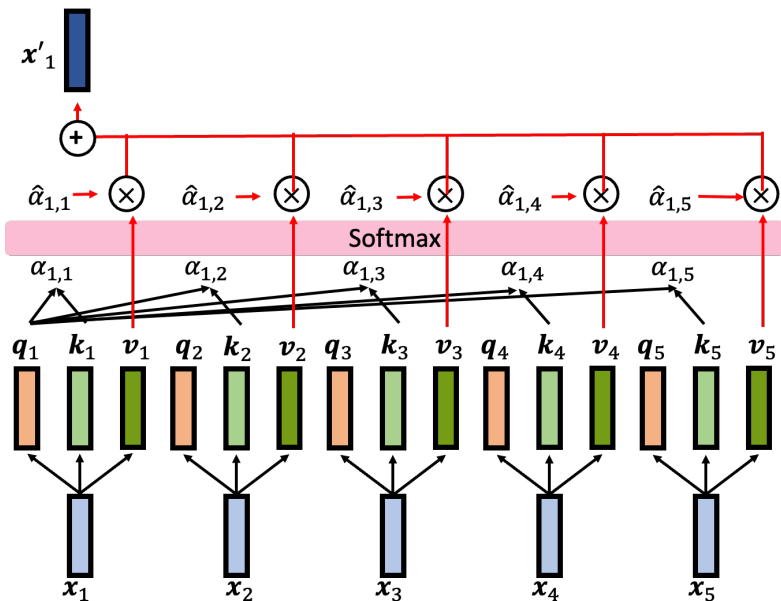


A Framework of Autoregressive LM

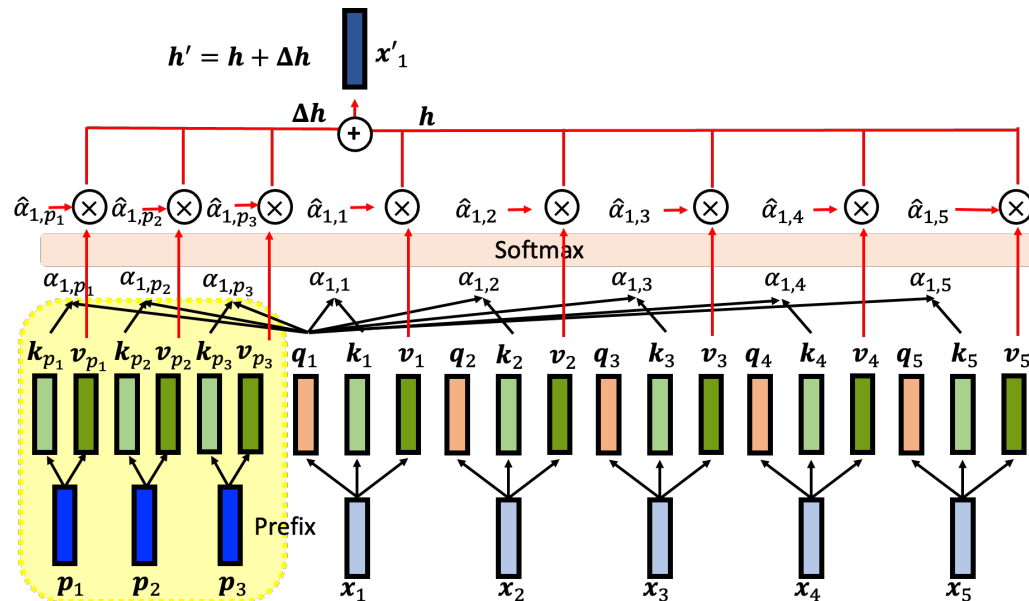
- **Continuous Prompt** (Soft Prompt): **Vectors** initialized from *word embeddings*
- **Prompt Tuning**: **Additional learnable parameters** injected into the model
- **Pros**: Learn **generalizable task-specific** embeddings and **parameter-efficient**
- **Cons**: **Hard to interpret** and cannot be applied to publicly unavailable models ^[1]

Part 2 – Continuous Prompt-based Learning

Prompt Tuning: **Additional learnable parameters** injected into the model



Original



Prefix Tuning

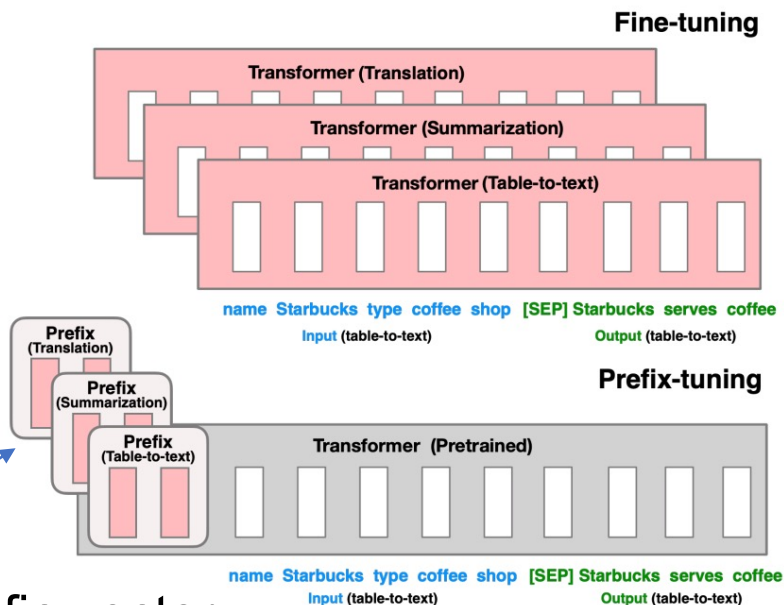
Part 2 – Continuous Prompt-based Learning

Prompt Tuning: **Additional learnable parameters** injected into the model

Prefix-Tuning

$$z = [\text{Prefix}; \mathbf{x}, \mathbf{y}]$$

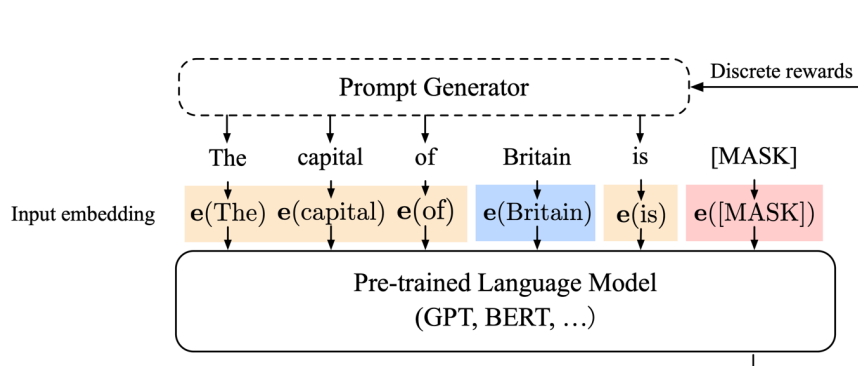
continuous task-specific vector



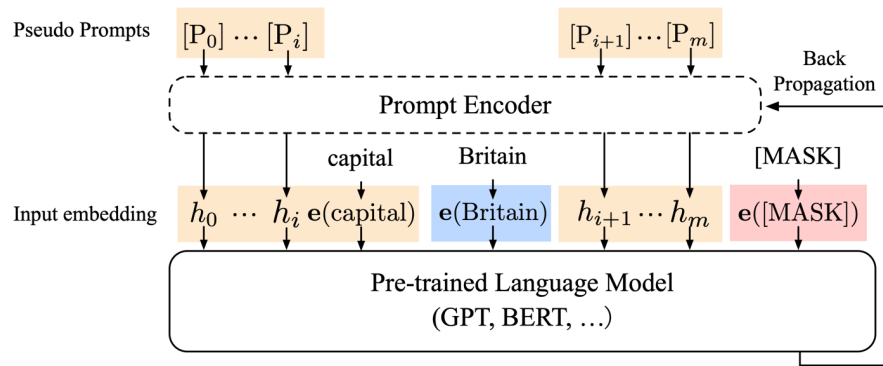
Part 2 – Continuous Prompt-based Learning

Prompt Tuning: **Additional learnable parameters** injected into the model

Pseudo-tuning (P-tuning)



(a) Discrete Prompt Search



(b) P-tuning

$$T = \{[\mathbf{P}_{0:i}], \mathbf{x}, [\mathbf{P}_{i+1:m}], \mathbf{y}\}$$

Part 2 – Continuous Prompt-based Learning

Prompt Tuning: **Additional learnable parameters** injected into the model

Soft Prompts

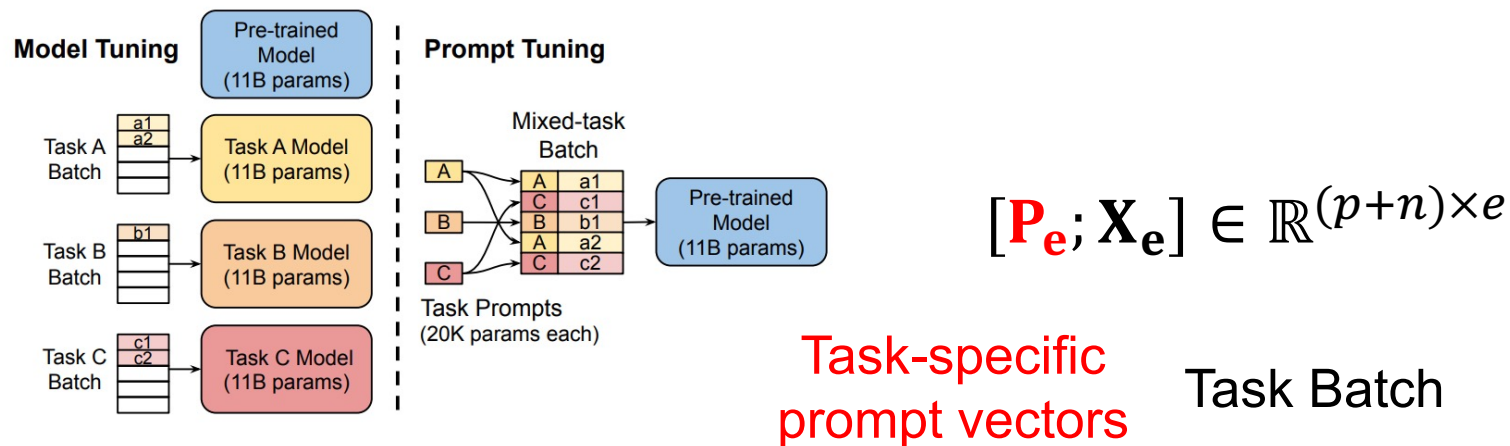
—x, **v₁**, **v₂**, **v₃**, ..., **v_n**, —y, **v_{n+1}**

$$\mathbf{v}_i^{(l)} \leftarrow \mathbf{v}_i^{(l)} + \Delta$$

Part 2 – Continuous Prompt-based Learning

Prompt Tuning: **Additional learnable parameters** injected into the model

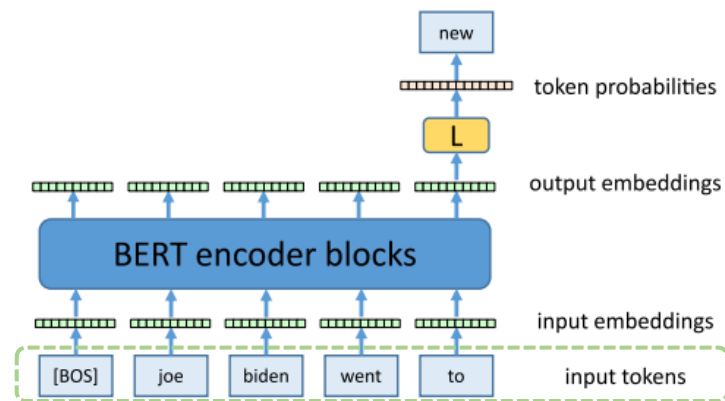
Prompt Tuning



Part 1 – Prompt Category – Discrete Prompt

Prompt	P@1
[X] is located in [Y]. (<i>original</i>)	31.29
[X] is located in which country or state? [Y].	19.78
[X] is located in which country? [Y].	31.40
[X] is located in which country? In [Y].	51.08

Table 1. Case study on LAMA-TREx P17 with bert-base-cased. A single-word change in prompts could yield a drastic difference.



A Framework of Autoregressive LM

- **Discrete Prompt** (Hard Prompt): **Words** that are originally *in the vocabulary*
- **Discrete Prompt Learning**: Learn lexical sequences
- **Pros**: **Easy to interpret**
- **Cons**: Require **domain expertise**/understanding, and **sub-optimal and sensitive** [1, 2]

Part 3 – Discrete Prompt-based Learning

Discrete Prompt-based Learning: Automatically learn prompts from training data

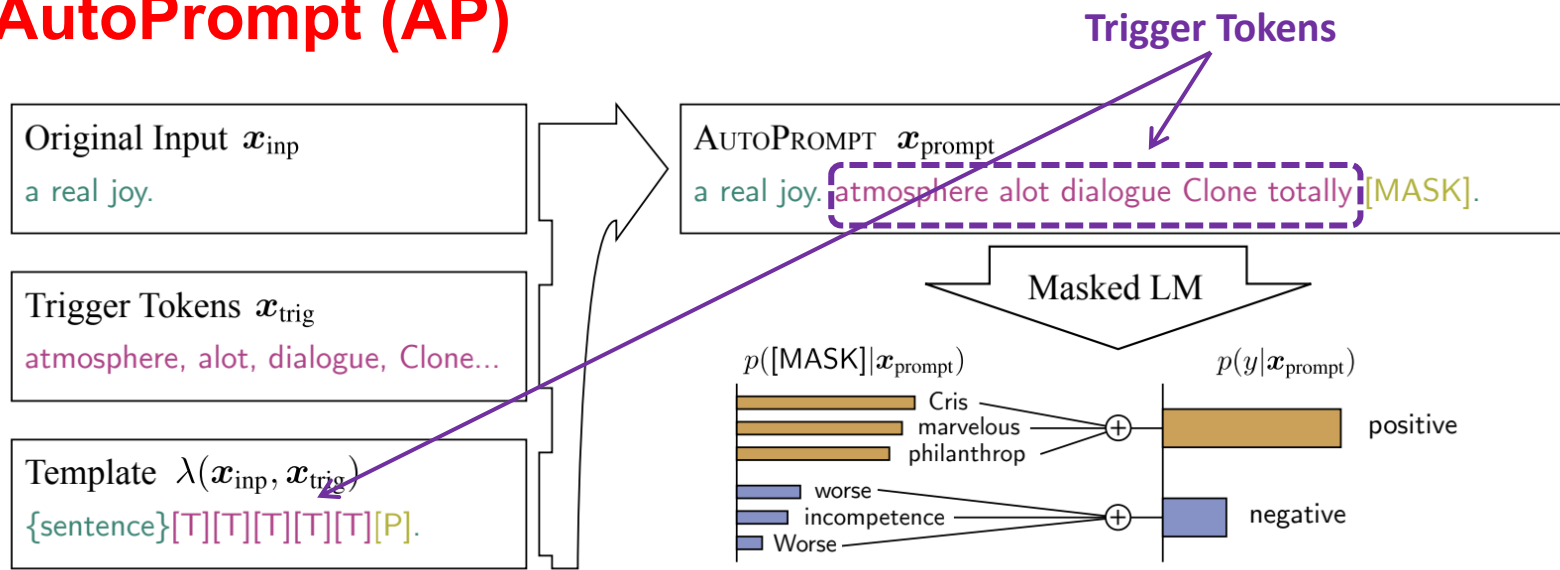
Relation	Method	Prompt	P@1
native-language-of (P103)	Manual	The native language of [X] is [Y]	74.54
	AP BERT	[X]PA communerug speaks proper [Y]	84.87
	AP RoBERTa	[X]neau optionally fluent!?!traditional [Y]	81.61
profession-of (P106)	Manual	[X] is a [Y] by profession	0.73
	AP BERT	[X] supporters studied politicians musician turned [Y]	15.83
	AP RoBERTa	[X] (), astronomers businessman·former [Y]	19.24
music-played-by (P136)	Manual	[X] plays [Y] music	0.7
	AP BERT	[X] freaking genre orchestra fiction acid [Y]	59.95
	AP RoBERTa	[X] blends postwar hostage drama sax [Y]	52.97

Table 1: Examples of prompts learnt by AP for the fact retrieval task for BERT and RoBERTa PLMs and the human-written manual prompts. T-REx relation ids are shown with brackets for each relation type. Precision@1 (P@1) scores are shown when each prompt is used in fact retrieval.

Part 3 – Discrete Prompt-based Learning

Discrete Prompt-based Learning: Automatically learn prompts from training data

AutoPrompt (AP)



AutoPrompt (AP) – Template

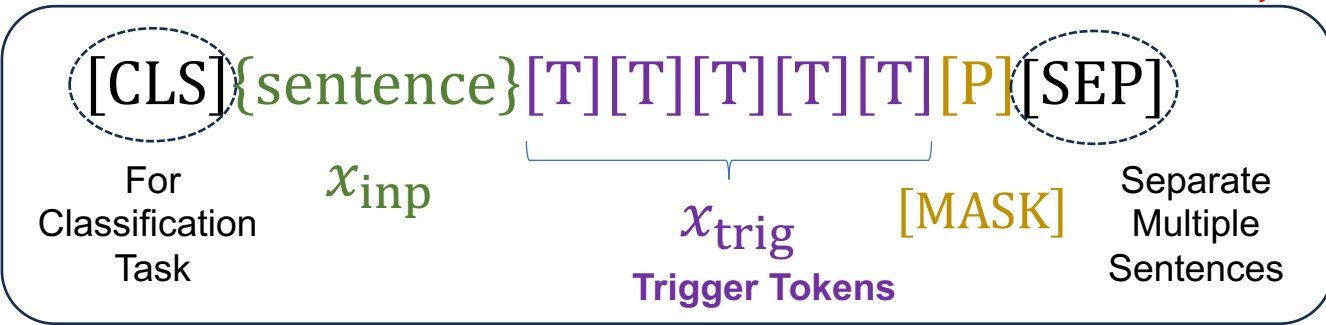
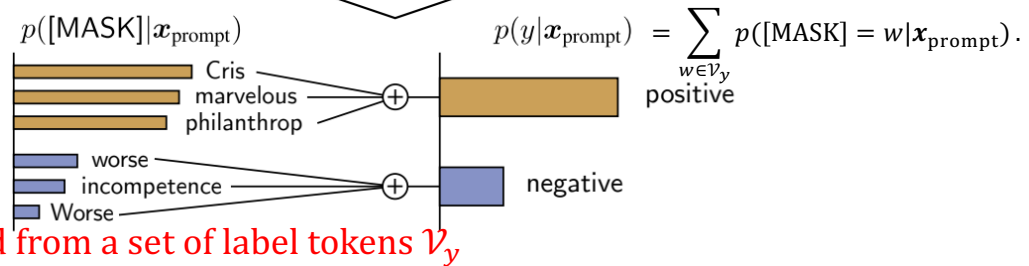
Original Input x_{inp}
a real joy.

Trigger Tokens x_{trig}
atmosphere, alot, dialogue, Clone...

Template $\lambda(x_{inp}, x_{trig})$
{sentence}[T][T][T][T][T][P].

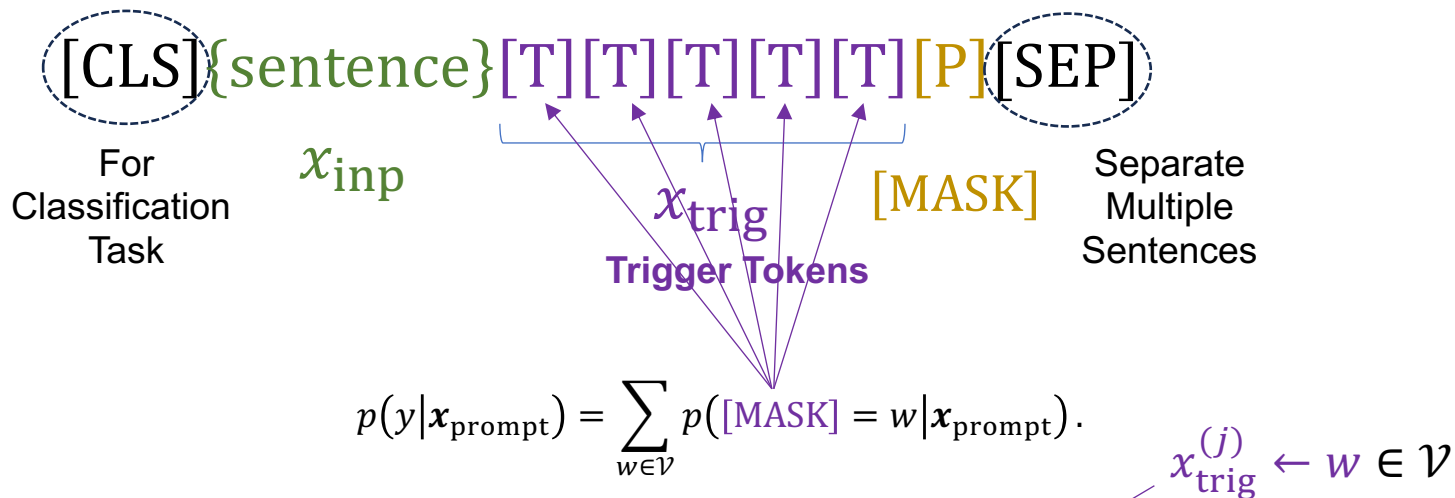
AUTOPROMPT x_{prompt}
a real joy. [atmosphere alot dialogue Clone totally][MASK].

Trigger Tokens



Relation
(sub, rel, obj)
Subject Object

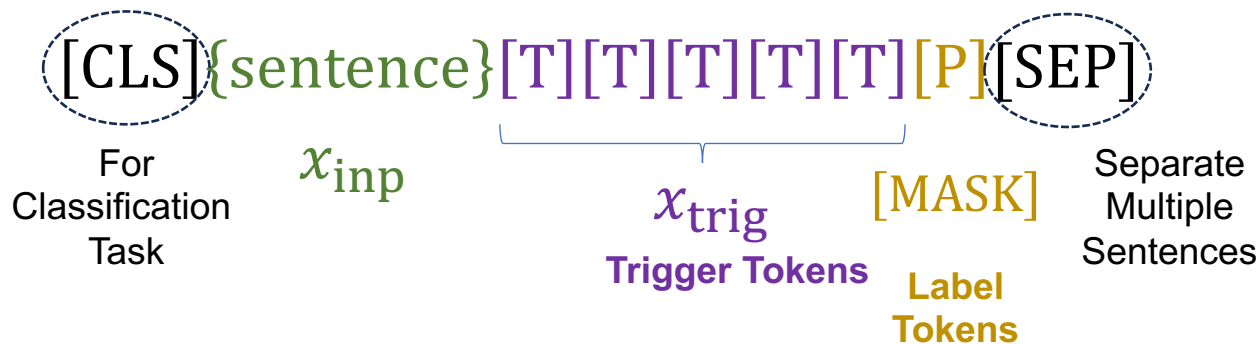
AutoPrompt (AP) – Gradient-based Prompt Search



First-order approximation to get a candidate set \mathcal{V}_{cand} :

$$\mathcal{V}_{cand} = \text{top-}k_{w \in \mathcal{V}} [\mathbf{w}_{in}^T \nabla \log p(y|x_{prompt})]. \quad \mathbf{W}_{in}: \text{input embedding of } w$$

AutoPrompt (AP) – Automating Label Token Selection



Classifier

$$\mathbf{h} = \text{Transformer}_{enc}(\tilde{\mathbf{x}}),$$

$$p(y|\mathbf{h}^{(i)}) \propto \exp(\mathbf{h}^{(i)} \cdot \mathbf{y} + \beta_y).$$

Words: $\mathbf{w}_{out} \cdot \mathbf{y}$
Labels: $\mathbf{h}^{(i)} \cdot \mathbf{y}$

$$\mathcal{V}_y = \text{top-}k_{w \in \mathcal{V}}[s(y, w)],$$

where score $s(y, w) = p(y|\mathbf{w}_{out}) \propto \exp(\mathbf{w}_{out} \cdot \mathbf{y} + \beta_y)$

Part 3 – Discrete Prompt-based Learning

Discrete Prompt-based Learning: automatically learn prompts from training data

Manually-written Prompts (MP)

{hypothesis}? | < MASK >, {premise}

Yes	• Entailment
No	• Contradiction
Maybe	• Neutral

Verbalizer
(injective function)

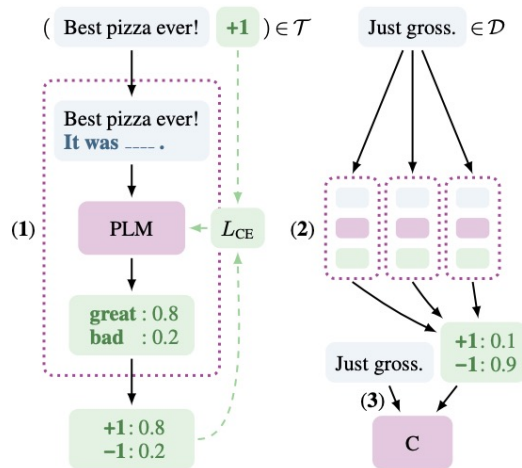


Figure 1: PET for sentiment classification. (1) A number of patterns encoding some form of task description are created to convert training examples to cloze questions; for each pattern, a pretrained language model is finetuned. (2) The ensemble of trained models annotates unlabeled data. (3) A classifier is trained on the resulting soft-labeled dataset.

Part 3 – Other Prompt-based Learning Methods

▪ Manual Prompt Design:

In-Context Learning – Brown *et al.*, Language Models are Few-Shot Learners, In NeurIPS'20.

Pattern Exploiting Training (PET) – Schick *et al.*, Exploiting Cloze Questions for Few Shot Text Classification and Natural Language Inference, In EACL'21.

▪ Mining and Paraphrasing-based Methods:

Jiang *et al.*, How Can We Know What Language Models Know?, In TACL'20.

▪ Gradient-based Search:

AutoPrompt – Shin *et al.*, AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts, In EMNLP'20.

▪ Automatic Prompt Generation:

Gao *et al.*, Making Pre-trained Language Models Better Few-shot Learners, In ACL'21.

Semantic preserving distance metric learning and applications



Jun Yu^a, Dapeng Tao^{b,*}, Jonathan Li^c, Jun Cheng^{d,e}

^a School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China

^b School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China

^c Department of Computer Science, Xiamen University, Xiamen, Fujian 361005, China

^d Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen, China

^e The Chinese University of HongKong, Shatin, HongKong, China

ARTICLE INFO

Article history:

Available online 3 February 2014

Keywords:

Clustering

Semantic distance metric

Pairwise constraints

Manifold structure

ABSTRACT

How do we accurately browse a large set of images or efficiently annotate the images from an image library? Image clustering methods are invaluable tools for applications such as content-based image retrieval and image annotation. To perform these tasks, it is critical to have proper features to describe the visual and semantic content of images and to define an **accurate distance metric to measure the dissimilarity between any two images**. However, existing methods, which adopt the features of color histograms, edge direction histograms and shape context, lack the ability to **describe semantic content**. To solve this problem, we propose a new approach that utilizes user-provided pairwise constraints to describe the **semantic relationship** between two images. A Semantic Preserving Distance Metric Learning (SP-DML) algorithm is developed to explore the complementary characteristics of the visual features and pairwise constraints in a unified feature space. In this space, the learned distance metric can be used to measure the dissimilarity between two images. Specifically, the manifold structure adopted in SP-DML is revealed by the image's visual features. To integrate semantic contents in distance metric learning, SP-DML utilizes pairwise constraints to build semantic patches and align these patches to obtain the optimal distance metric for the new feature space. Experimental results in image clustering demonstrate that the performance of SP-DML is appealing.

Can we propose a Metric to measure the semantic preserving distance between two prompts?

Part 4 – Prompt Perturbation and Robustness

- **Prompt Perturbation:** alter or modify the original input prompt or query to generate different or varied responses.

Question: Is **semantic-preserving distance (maybe measuring the distance between deep features)** essential to prompt perturbation?

define



- **Certified Robustness:**

The model $f(\cdot)$ is certified robust if it satisfies the following condition for $\forall x$

$$f(\mathbf{x}') = y,$$

$$\|\mathbf{x}' - \mathbf{x}\|_0 = \sum_{i=1}^L \mathbb{I}(x'_i \neq x_i) \leq dL.$$

$\mathbf{x} = [x_1, x_2, \dots, x_L]$: input to the LLM $f(\cdot)$

$\|\mathbf{x}' - \mathbf{x}\|_0$: Hamming Distance

$\mathbb{I}(\cdot)$: Indicator Function

d : perturbation scale

Part 4 – Prompt Perturbation and Robustness



Could you please give a comprehensive list of Prompt Perturbation of LLM input?



Certainly, here's a comprehensive list of various techniques for perturbing LLM (Language Model) input prompts:



1. **Rephrasing:** Reformulate the prompt with synonymous phrases or alternative wording to elicit different responses.
2. **Context Addition:** Include additional context or background information to provide the model with more context to generate a response.
3. **Context Removal:** Remove portions of the prompt or context to see how the model performs with limited information.
4. **Context Expansion:** Expand on the prompt by adding more details or examples to encourage the model to provide a more detailed response.
5. **Context Shift:** Change the context of the prompt to explore how the model responds to different scenarios or perspectives.
6. **Parameter Tweaking:** Adjust the model's parameters or hyperparameters, such as temperature or max tokens, to control the randomness or length of responses.
7. **Seed Text:** Start the prompt with a specific seed text or sentence to guide the model's response in a particular direction.
8. **Question Transformation:** Change a statement into a question or vice versa to explore the model handles different types of queries.

Regenerate

Part 5 – Prompt Robustness Evaluation

- **Rate of Degradation (RoD):** the decrease in accuracy of the target task due to the perturbations added to the prompt.
- A smaller RoD indicates a more robust model against perturbations

$$\text{RoD} = \frac{\text{avgacc}_x - \text{avgacc}_{x^*}}{\text{avgacc}_x} = 1 - \frac{\text{avgacc}_{x^*}}{\text{avgacc}_x},$$

- where x^* is the perturbed version of the original prompt x , and avgacc_x and avgacc_{x^*} are the averaged accuracies over M prompts

Credits:

Meyers *et al.*, Signal Processing on PV Time-series Data: Robust Degradation Analysis Without Physical Models, In IEEE Journal of Photovoltaics'19.
Ishibashi *et al.*, Evaluating the Robustness of Discrete Prompts, In EACL'23.

Thank you very much for your attention!

Dependable Computing Laboratory,
Department of Electrical and Computer Engineering,
Boston University

