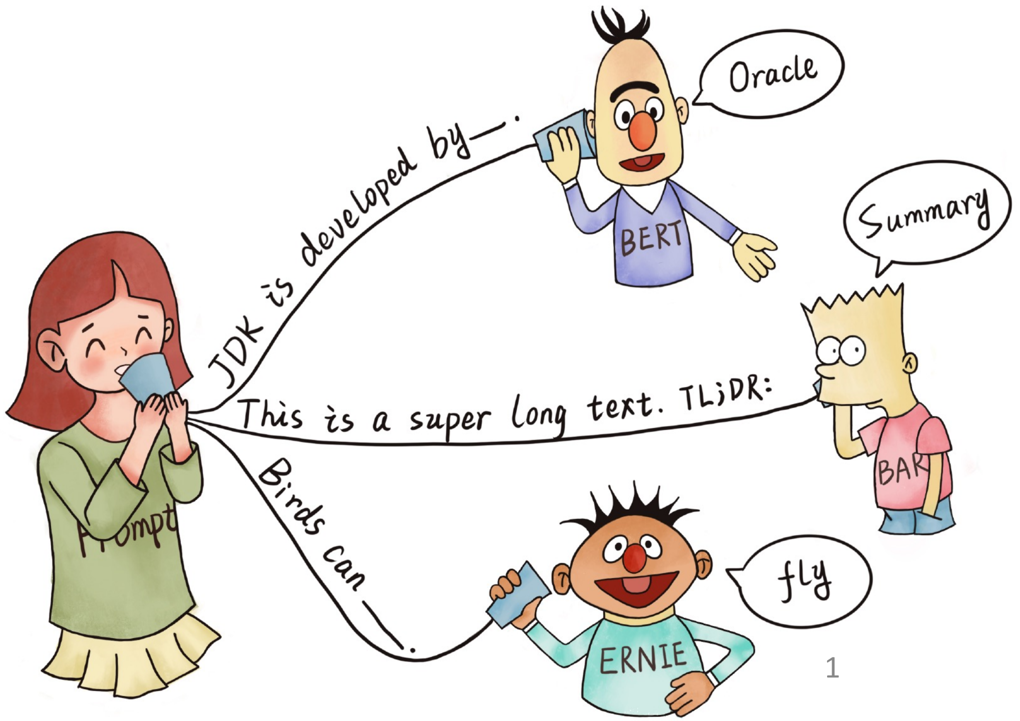
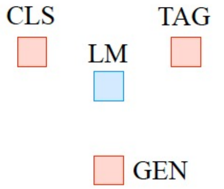
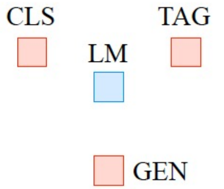
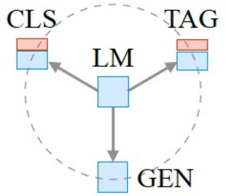
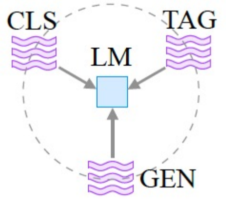
# Prompting Methods in NLP

Qintong Li

2021/08/05



# Four Paradigm in NLP

| Paradigm | Engineering | Task Relation |
|---|---|---|
| a. Fully Supervised Learning (Non-Neural Network) | Features (e.g. word identity, part-of-speech, sentence length) |   |
| b. Fully Supervised Learning (Neural Network) | Architecture (e.g. convolutional, recurrent, self-attentional) |   |
| c. Pre-train, Fine-tune | Objective (e.g. masked language modeling, next sentence prediction) |   |
| d. Pre-train, Prompt, Predict | Prompt (e.g. cloze, prefix) |   |

- Traditional supervised learning (using supervised dataset):

$$P(y|x;\theta)$$

- Prompt-based learning:

$$P(x;\theta)$$

  - Prompt addition: $x' = f_{prompt}(x)$ [X] Overall, it was a [Z] movie.

    "I love this movie. Overall it was a [Z] movie."

  - Answer search: $f_{fill}(x', z)$;

    I love this movie. Overall, it was a bad movie.

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

    I love this movie. Overall, it was a good movie.

  - Answer mapping: $\hat{z} \rightarrow y$ the answer itself is the output

    (e.g. "excellent", "fabulous", "wonderful") to represent a single class (e.g. "++").

Two advantages:

1. It allows the language model to be pre-trained on massive amounts of raw text;
2. By defining a new prompting function the model is able to perform few-shot or even zero-shot learning.

# Design Considerations for Prompting

- Pre-trained Model Choice $P(x;\theta)$
- Prompt Engineering $f_{prompt}$
- Answer Engineering design $\mathcal{Z}$
- Expanding the Paradigm e.g., use of multiple prompts
- Prompt-based Training Strategies LM, Prompt

# Design Considerations for Prompting

- Pre-trained Model Choice $P(x;\theta)$
- Prompt Engineering $f_{prompt}$
- Answer Engineering
- Expanding the Paradigm e.g., use of multiple prompts
- Prompt-based Training Strategies LM, Prompt

# Prompt Engineering $f_{\text{prompt}}(\boldsymbol{x})$

- Prompt shape
  - Cloze prompts: fill in the blanks of a textual string
  - Prefix prompts: continue a string prefix
- Prompt acquisition
  - Manual template engineering
  - Automated template learning
    - Discrete prompts: the prompt is an actual text string
    - Continuous prompts: the prompt is instead decribed in the embedding space of the underlying LM

# Discrete Prompts

## AUTOPROMPT: Eliciting Knowledge from Language Models with Automatically Generated Prompts

Taylor Shin[*◇] Yasaman Razeghi[*◇] Robert L. Logan IV[*◇]
Eric Wallace[♠] Sameer Singh[◇]
[◇]University of California, Irvine [♠]University of California, Berkeley
{tshin1, yrazeghi, rlogan, sameer}@uci.edu
ericwallace@berkeley.edu

EMNLP 2020

# Motivations

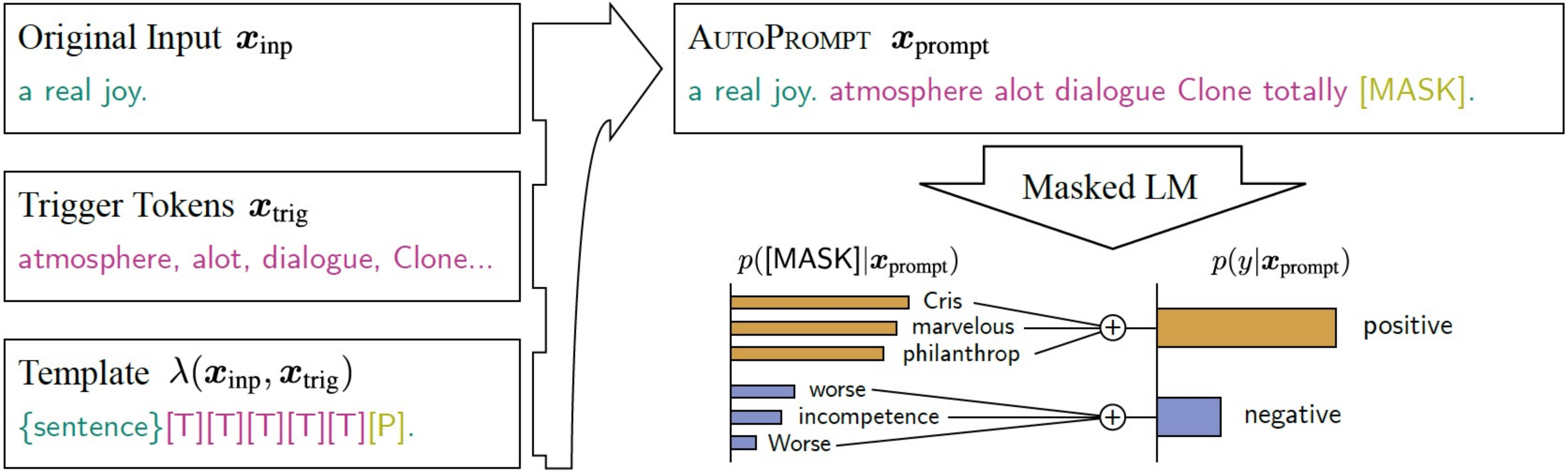
How can we directly evaluate the knowledge present in pretrained LMs, be it linguistic, factual, commonsense, or task-specific?

- Probing classifiers
    - false positives
    - high probing accuracy is not a sufficient condition
- Attention visualization
    - attention scores may be correlated with, but not caused by the underlying target knowledge
- Prompting
    - Provide a lower bound on what the language model “knows”

# Contributions

AUTOPROMPT—an automated method for generating prompts: Given a task, e.g., sentiment analysis, AUTOPROMPT creates a prompt by combining the original task inputs (e.g. reviews) with a collection of trigger tokens according to a template.

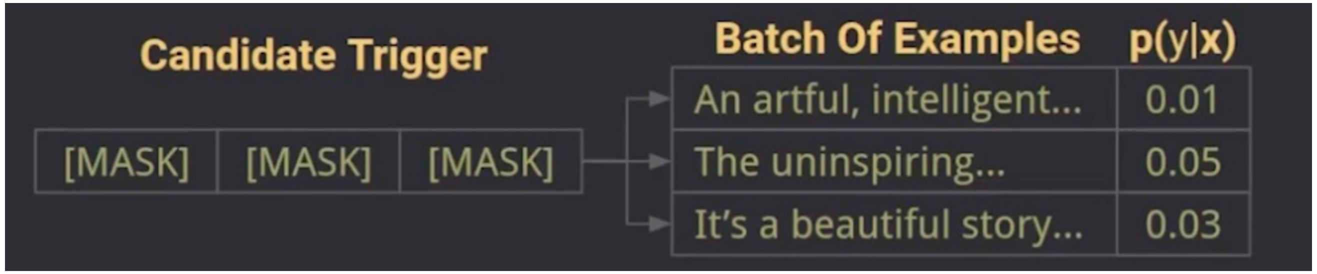
- Parameter-free
- Use output instead of internal representation
- Single model can solve many tasks
- No need manual effort
- Better than hand written and sometimes better than fine-tuning


Original Input $\boldsymbol{x}_{\text{inp}}$
a real joy.
Trigger Tokens $\boldsymbol{x}_{\text{trig}}$
atmosphere, alot, dialogue, Clone...
Template $\lambda(\boldsymbol{x}_{\text{inp}}, \boldsymbol{x}_{\text{trig}})$
{sentence}[T][T][T][T][T][P].
AutoPrompt $\boldsymbol{x}_{\text{prompt}}$
a real joy. atmosphere alot dialogue Clone totally [MASK].
Masked LM
$p([\text{MASK}]|\boldsymbol{x}_{\text{prompt}})$
$p(y|\boldsymbol{x}_{\text{prompt}})$
Cris
marvelous
philanthrop
worse
incompetence
Worse
positive
negative

# Trigger Search

{sentence}[T][T][T][T][T][P].

The idea is to add a number of "trigger" tokens that are shared across all prompts (denoted by [T]) to help model predict the correct label .

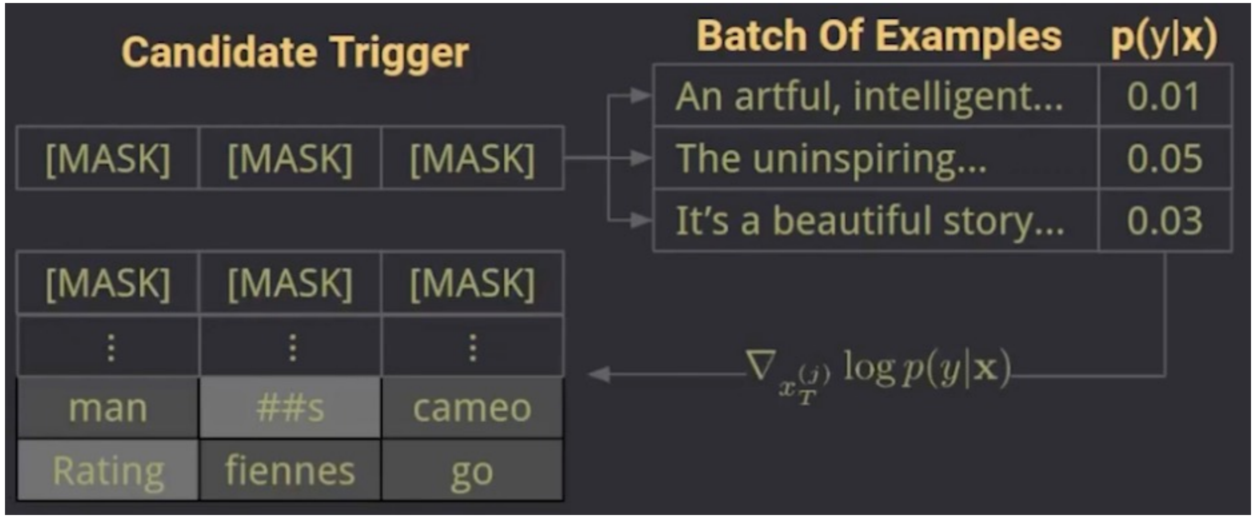



Wallace et al. Universal Adversarial Triggers for Attacking and Analyzing NLP.

# Trigger Search

{sentence}[T][T][T][T][T][P].

The idea is to add a number of “trigger” tokens that are shared across all prompts (denoted by [T]) to help model predict the correct label .



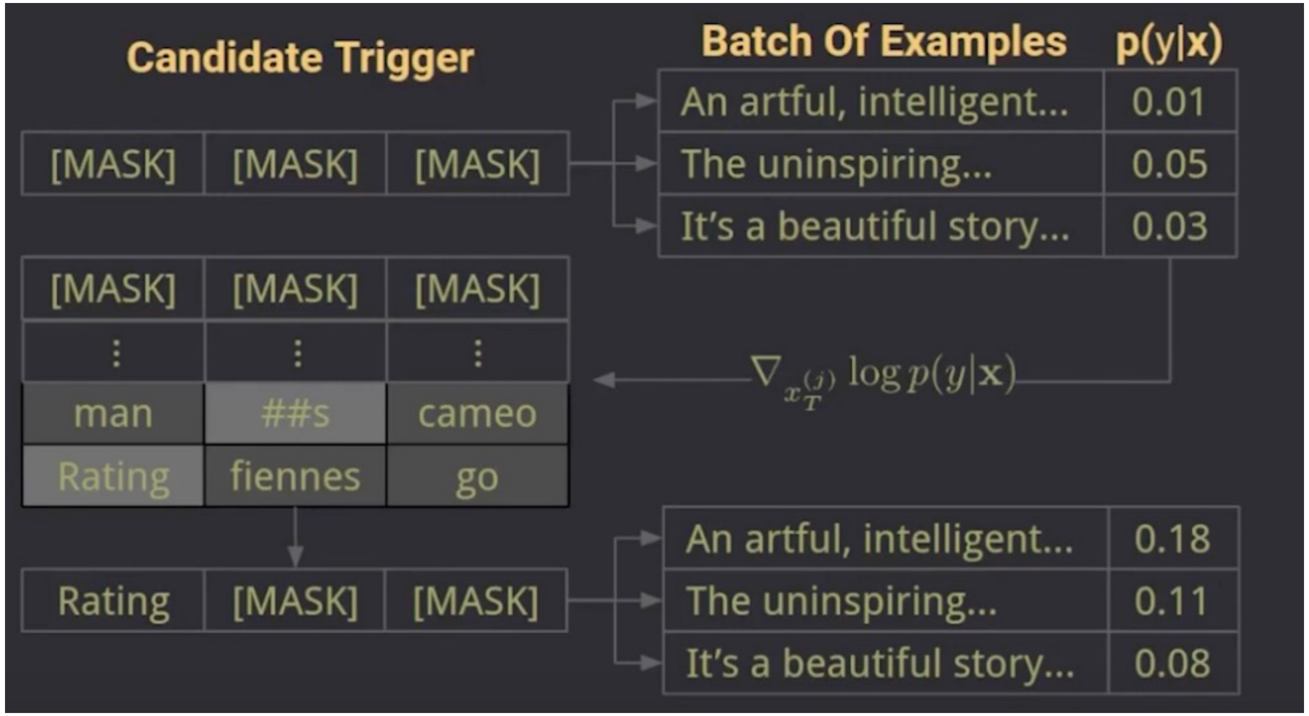

$$p(y|\boldsymbol{x}_{\text{prompt}}) = \sum_{w \in \mathcal{V}_y} p([\text{MASK}] = w|\boldsymbol{x}_{\text{prompt}})$$

Wallace et al. Universal Adversarial Triggers for Attacking and Analyzing NLP.

# Trigger Search

{sentence}[T][T][T][T][T][P].

The idea is to add a number of "trigger" tokens that are shared across all prompts (denoted by [T]) to help model predict the correct label .



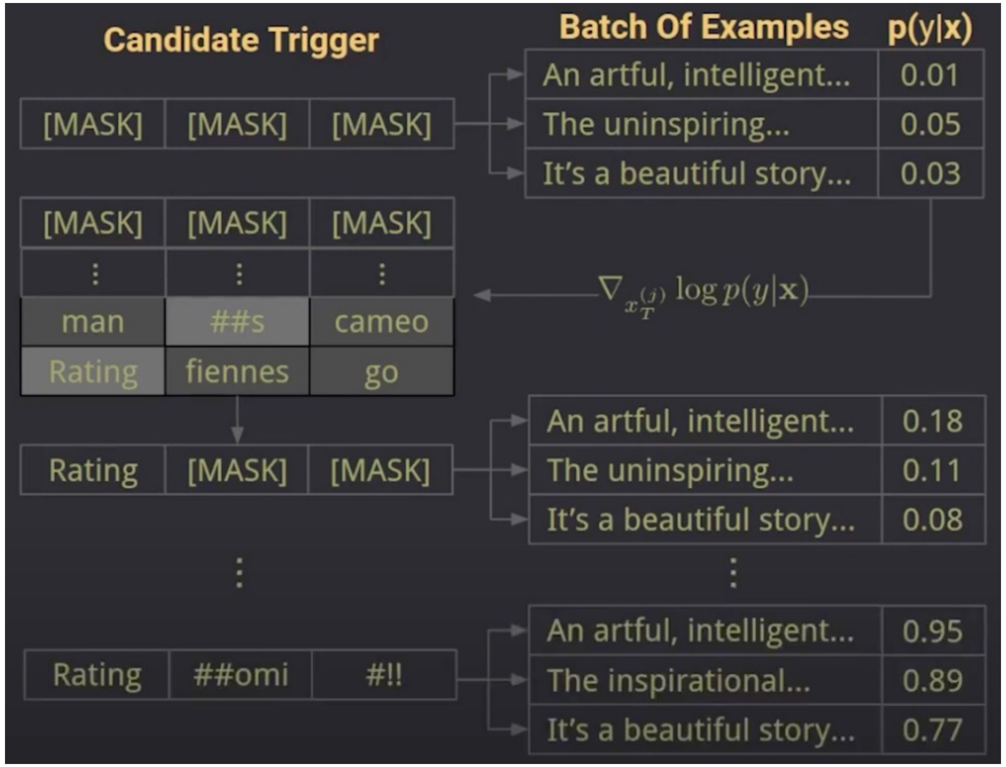

$$\mathcal{V}_{\text{cand}} = \underset{w \in \mathcal{V}}{\text{top-}k} \left[ \boldsymbol{w}_{\text{in}}^T \nabla \log p(y|\boldsymbol{x}_{\text{prompt}}) \right]$$

Wallace et al. Universal Adversarial Triggers for Attacking and Analyzing NLP.

# Trigger Search

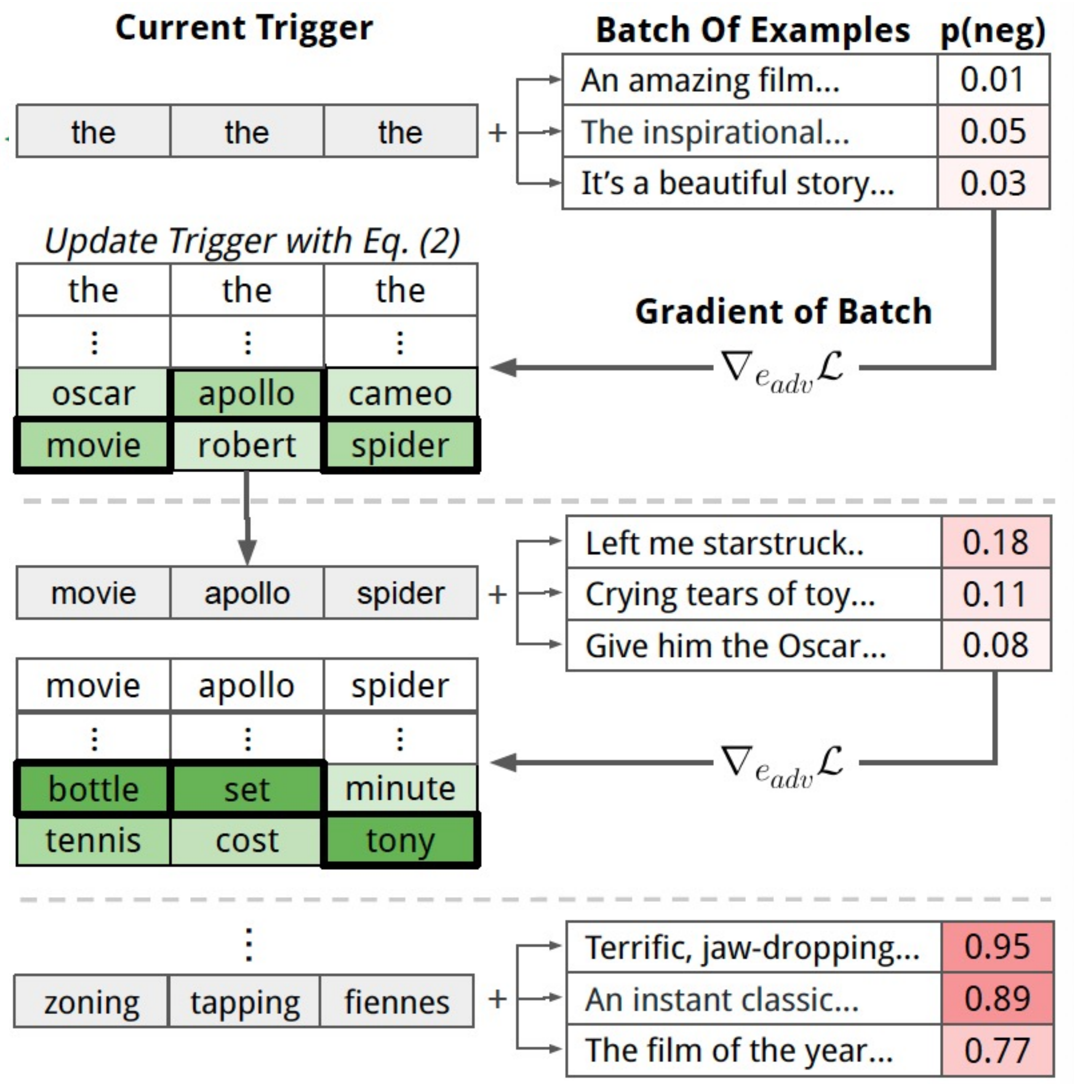
{sentence}[T][T][T][T][T][P].

The idea is to add a number of “trigger” tokens that are shared across all prompts (denoted by [T]) to help model predict the correct label .




Wallace et al. Universal Adversarial Triggers for Attacking and Analyzing NLP.

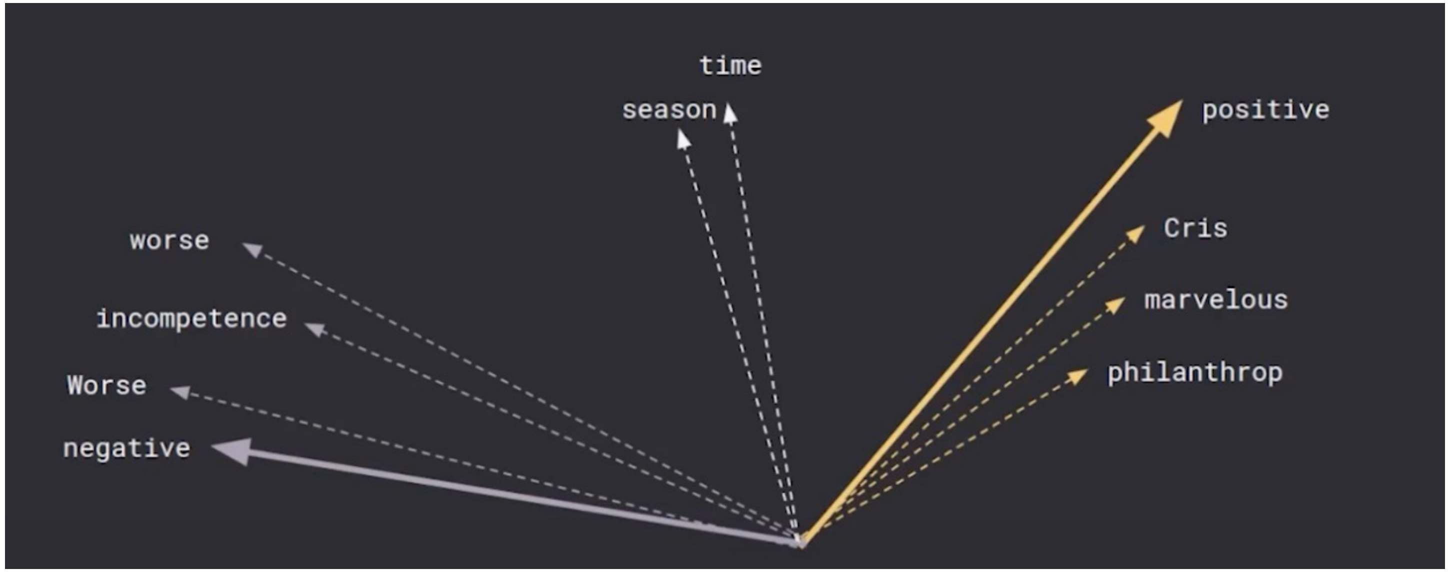
# Trigger Search




Wallace et al. Universal Adversarial Triggers for Attacking and Analyzing NLP.

# Label Tokens

- Train a logistic classifier to predict the class label using the contextualized embedding of the [MASK] token as input.
- Find the words whose vector representations in the output layer of the language model are closest to the learned representations of these labels.



# Experiments

| Task | Prompt Template | Prompt found by AUTOPROMPT | Label Tokens |
|---|---|---|---|
| Sentiment Analysis | {sentence} [T]. . . [T] [P]. | unflinchingly bleak and desperate Writing academicswhere overseas will appear [MASK]. | **pos**: partnership, extraordinary, ##bla<br>**neg**: worse, persisted, unconstitutional |
| NLI | {prem}[P][T]. . . [T]{hyp} | Two dogs are wrestling and hugging [MASK] concretepathic workplace There is no dog wrestling and hugging | **con**: Nobody, nobody, nor<br>**ent**: ##found, ##ways, Agency<br>**neu**: ##ponents, ##lary, ##uated |
| Fact Retrieval | *X plays Y music*<br>{sub}[T]. . . [T][P]. | Hall Overton fireplacemade antique son alto [MASK]. | |
| Relation Extraction | *X is a Y by profession*<br>{sent}{sub}[T]. . . [T][P]. | Leonard Wood (born February 4, 1942) is a former Canadian politician.<br>Leonard Wood gymnasium brotherdicative himself another [MASK]. | |

Table 3: **Example Prompts** by AUTOPROMPT for each task. On the left, we show the prompt template, which combines the input, a number of trigger tokens [T], and a prediction token [P]. For classification tasks (sentiment analysis and NLI), we make predictions by summing the model's probability for a number of automatically selected label tokens. For fact retrieval and relation extraction, we take the most likely token predicted by the model.

# Experiments

## Manual prompt VS AutoPrompt

| Model | Dev | Test |
|---|---|---|
| BiLSTM | - | 82.8† |
| BiLSTM + ELMo | - | 89.3† |
| BERT (linear probing) | 85.2 | 83.4 |
| BERT (finetuned) | - | 93.5† |
| RoBERTa (linear probing) | 87.9 | 88.8 |
| RoBERTa (finetuned) | - | 96.7† |
| BERT (manual) | 63.2 | 63.2 |
| BERT (AUTOPROMPT) | 80.9 | 82.3 |
| RoBERTa (manual) | 85.3 | 85.2 |
| RoBERTa (AUTOPROMPT) | 91.2 | 91.4 |

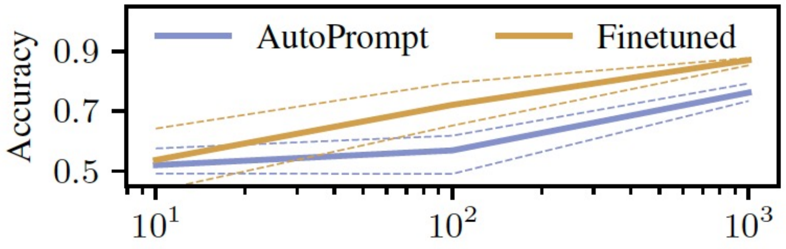
Table 1: **Sentiment Analysis** performance on the SST-2 test set of supervised classifiers (top) and fill-in-the-blank MLMs (bottom). Scores marked with † are from the GLUE leaderboard: http://gluebenchmark.com/leaderboard.

| Prompt Type | Original | | | T-REx | | |
|---|---|---|---|---|---|---|
| | MRR | P@10 | P@1 | MRR | P@10 | P@1 |
| LAMA | 40.27 | 59.49 | 31.10 | 35.79 | 54.29 | 26.38 |
| LPAQA (Top1) | 43.57 | 62.03 | 34.10 | 39.86 | 57.27 | 31.16 |
| AUTOPROMPT 5 Tokens | 53.06 | 72.17 | 42.94 | 54.42 | 70.80 | 45.40 |
| AUTOPROMPT 7 Tokens | 53.89 | 73.93 | 43.34 | 54.89 | 72.02 | 45.57 |

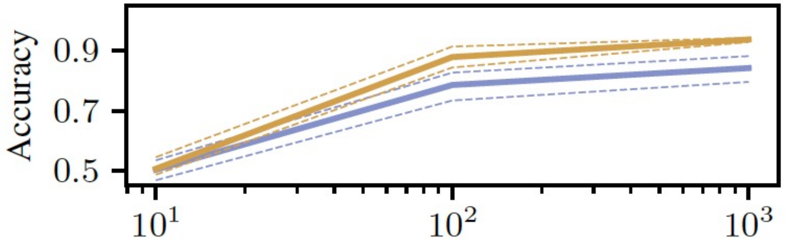
Factual Retrieval

# Discussion

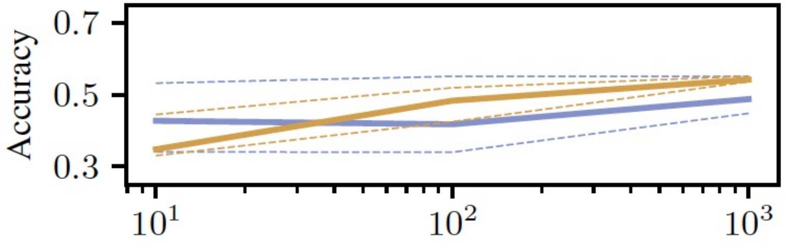
- Prompting as an Alternative to Finetuning
  - Prompts generated using AUTOPROMPT can achieve higher accuracy than finetuning in the low-data regime.
  - Prompting has advantages over finetuning when trying to solve many different tasks.



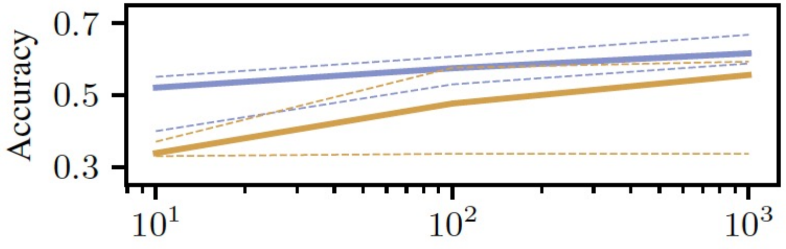

(a) BERT on SST-2




(b) RoBERTa on SST-2




(c) BERT on SICK-E




(d) RoBERTa on SICK-E

# Discussion

- Limitations of Prompting
  - Preliminary evaluation on datasets such as QQP (Iyer et al., 2017) and RTE (Dagan et al., 2005), prompts generated manually and with AUTOPROMPT did not perform considerably better than chance.
  - AUTOPROMPT makes prompt-based probes more generally applicable, but, it still remains just one tool in the toolbox of the interpretability researcher.

# Discussion

- Limitations of AUTOPROMPT
  - It requires labeled training data.
  - AUTOPROMPT generated prompts lack interpretability.
  - It can sometimes struggle when the training data is highly imbalanced.
  - The large discrete space of phrases (need more more effective crafting techniques).

| Model | SICK-E Datasets standard | 3-way | 2-way |
|---|---|---|---|
| Majority | 56.7 | 33.3 | 50.0 |
| BERT (finetuned) | 86.7 | 84.0 | 95.6 |
| BERT (linear probing) | 68.0 | 49.5 | 91.9 |
| RoBERTa (linear probing) | 72.6 | 49.4 | 91.1 |
| BERT (AUTOPROMPT) | 62.3 | 55.4 | 85.7 |
| RoBERTa (AUTOPROMPT) | 65.0 | 69.3 | 87.3 |

# Prompt-based Training Strategies

| Strategy | LM Params | Prompt Params | | Example |
|---|---|---|---|---|
| | | Additional | Tuned | |
| Promptless Fine-tuning | Tuned | - | | ELMo [130], BERT [32], BART [94] |
| Tuning-free Prompting | Frozen | ✗ | ✗ | GPT-3 [16], AutoPrompt [159], LAMA [133] |
| Fixed-LM Prompt Tuning | Frozen | ✓ | Tuned | Prefix-Tuning [96], Prompt-Tuning [91] |
| Fixed-prompt LM Tuning | Tuned | ✗ | ✗ | PET-TC [153], PET-Gen [152], LM-BFF [46] |
| Prompt+LM Fine-tuning | Tuned | ✓ | Tuned | PADA [8], P-Tuning [103], PTR [56] |

Table 6: Characteristics of different tuning strategies. "Additional" represents if there are additional parameters beyond LM parameters while "Tuned" denotes if parameters are updated.

# Prefix-Tuning: Optimizing Continuous Prompts for Generation

**Xiang Lisa Li**
Stanford University
xlisali@stanford.edu

**Percy Liang**
Stanford University
pliang@cs.stanford.edu

# Overview

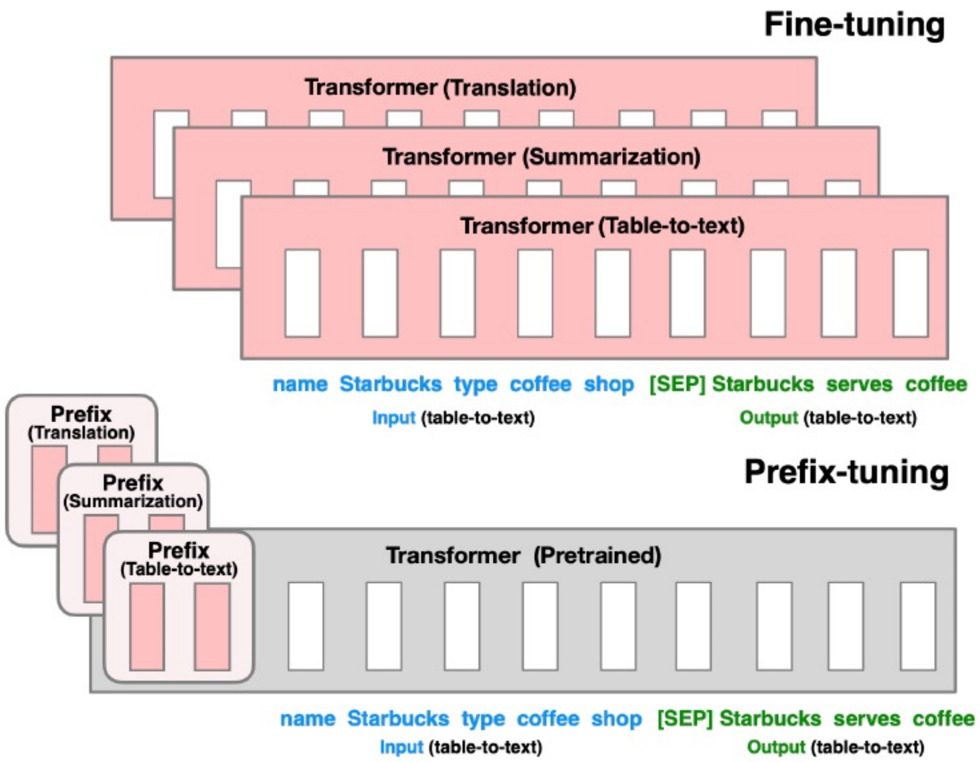



Figure 1: Fine-tuning (top) updates all Transformer parameters (the red Transformer box) and requires storing a full model copy for each task. We propose prefix-tuning (bottom), which freezes the Transformer parameters and only optimizes the prefix (the red prefix blocks). Consequently, we only need to store the prefix for each task, making prefix-tuning modular and space-efficient. Note that each vertical block denote transformer activations at one time step.

# Method

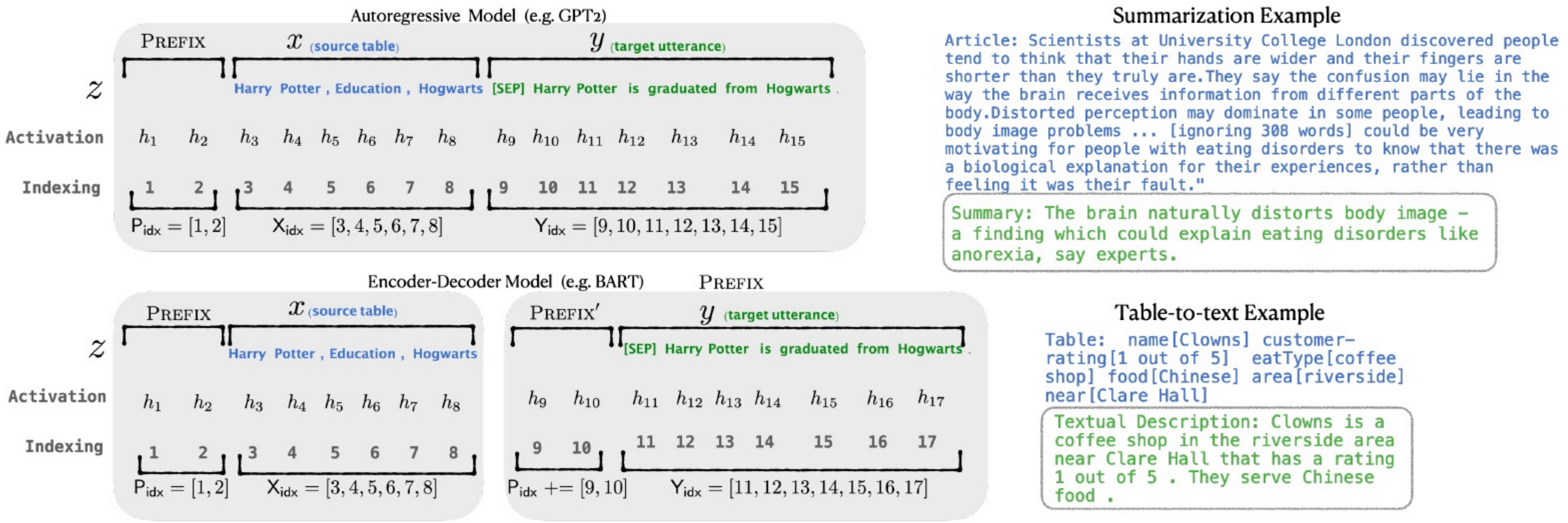



Figure 2: An annotated example of prefix-tuning using an autoregressive LM (top) and an encoder-decoder model (bottom). The prefix activations $\forall i \in \mathsf{P}_{\text{idx}}, h_i$ are drawn from a trainable matrix $P_\theta$. The remaining activations are computed by the Transformer.

# Experiments

| | E2E | | | | | WebNLG | | | | | | | | | DART | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | BLEU | NIST | MET | R-L | CIDEr | BLEU | | | MET | | | TER ↓ | | | BLEU | MET | TER ↓ | Mover | BERT | BLEURT |
| | | | | | | S | U | A | S | U | A | S | U | A | | | | | | |
| | | | | | | | | | GPT-2$_{MEDIUM}$ | | | | | | | | | | | |
| FINE-TUNE | 68.2 | 8.62 | **46.2** | 71.0 | 2.47 | **64.2** | 27.7 | 46.5 | **0.45** | 0.30 | 0.38 | **0.33** | 0.76 | 0.53 | 46.2 | **0.39** | **0.46** | **0.50** | **0.94** | **0.39** |
| FT-TOP2 | 68.1 | 8.59 | 46.0 | 70.8 | 2.41 | 53.6 | 18.9 | 36.0 | 0.38 | 0.23 | 0.31 | 0.49 | 0.99 | 0.72 | 41.0 | 0.34 | 0.56 | 0.43 | 0.93 | 0.21 |
| ADAPTER(3%) | 68.9 | 8.71 | 46.1 | 71.3 | 2.47 | 60.4 | **48.3** | 54.9 | 0.43 | **0.38** | **0.41** | 0.35 | **0.45** | **0.39** | 45.2 | 0.38 | **0.46** | **0.50** | **0.94** | **0.39** |
| ADAPTER(0.1%) | 66.3 | 8.41 | 45.0 | 69.8 | 2.40 | 54.5 | 45.1 | 50.2 | 0.39 | 0.36 | 0.38 | 0.40 | 0.46 | 0.43 | 42.4 | 0.36 | 0.48 | 0.47 | **0.94** | 0.33 |
| PREFIX(0.1%) | **69.7** | **8.81** | 46.1 | **71.4** | **2.49** | 62.9 | 45.6 | **55.1** | 0.44 | **0.38** | **0.41** | 0.35 | 0.49 | 0.41 | **46.4** | 0.38 | **0.46** | **0.50** | **0.94** | **0.39** |
| | | | | | | | | | GPT-2$_{LARGE}$ | | | | | | | | | | | |
| FINE-TUNE | 68.5 | 8.78 | 46.0 | 69.9 | 2.45 | **65.3** | 43.1 | 55.5 | **0.46** | 0.38 | **0.42** | **0.33** | 0.53 | 0.42 | **47.0** | **0.39** | 0.46 | **0.51** | **0.94** | **0.40** |
| Prefix | **70.3** | **8.85** | **46.2** | **71.7** | **2.47** | 63.4 | **47.7** | **56.3** | 0.45 | **0.39** | **0.42** | 0.34 | **0.48** | **0.40** | 46.7 | **0.39** | **0.45** | **0.51** | **0.94** | **0.40** |
| SOTA | 68.6 | 8.70 | 45.3 | 70.8 | 2.37 | 63.9 | 52.8 | 57.1 | 0.46 | 0.41 | 0.44 | - | - | - | - | - | - | - | - | - |

| | R-1 ↑ | R-2 ↑ | R-L ↑ |
|---|---|---|---|
| FINE-TUNE(Lewis et al., 2020) | 45.14 | 22.27 | 37.25 |
| PREFIX(2%) | 43.80 | 20.93 | 36.05 |
| PREFIX(0.1%) | 42.92 | 20.03 | 35.05 |

# Discussion

- Prefix-tuning, a lightweight alternative to fine-tuning that prepends a trainable continuous prefix for NLG tasks.
- Despite learning 1000x fewer parameters than finetuning, prefix-tuning can maintain a comparable performance in a full data setting and outperforms fine-tuning in both low-data and extrapolation settings.