# Making Pre-trained Language Models Better Few-shot Learners

Tianyu Gao, Adam Fisch, Danqi Chen
(Princeton & MIT)

# In this paper

Tasks:

- single sentence classification, e.g sentiment classification, grammar error prediction
- Sentence pair classification: NLI, paraphrase
- Sentence pair regression: sentence similarity

Method: Prompt-based Few-shot learning based on GPT

What to learn:

- Finetune method for LMs as FSL
- Automatic Prompt Generation using LM
- Incorporate demonstration into template

# Prompt based Finetuning

Given an input sentence x1=”No reason to watch it”

$$x_{\text{prompt}} = \texttt{[CLS]}\, x_1 \text{ It was } \texttt{[MASK]}\texttt{.}\ \texttt{[SEP]}$$

The LMs predicts whether the [MASK] to be **great (positive)** or **terrible (negative)**

# Automatic selection of label words

For each class **c**, construct a pruned set V^c of the top k word based on their conditional likelihood.

$$\underset{v \in \mathcal{V}}{\text{Top-}k} \left\{ \sum_{x_{\text{in}} \in \mathcal{D}^{c}_{\text{train}}} \log P_{\mathcal{L}}\Big( \texttt{[MASK]} = v \mid \mathcal{T}(x_{\text{in}}) \Big) \right\}, \quad (4)$$

The ranking of assignment before finetuning does not preserve after finetuning. This paper use dev set to rerank the pruned set.

# Automatic generation of templates

Target: generate a large set of templates based on a fixed set of labeled words.

This paper use T5 model to generate because T5 is trained on the same training signal.

*“Thank you <X> me to your party <Y> week”*.

*“<X> for inviting <Y> last <Z>”*.

Use T5 to generate template, using the label words

$$
\begin{aligned}
<S_1> &\longrightarrow \ <\mathrm{X}>\ \mathcal{M}(y)\ <\mathrm{Y}>\ <S_1> \\
<S_1> &\longrightarrow \ <S_1>\ <\mathrm{X}>\ \mathcal{M}(y)\ <\mathrm{Y}> \\
<S_1>, <S_2> &\longrightarrow \ <S_1>\ <\mathrm{X}>\ \mathcal{M}(y)\ <\mathrm{Y}>\ <S_2>
\end{aligned}
$$

# Automatic generation of templates

Use T5 to generate template, using the label words

$$<S_1> \;\longrightarrow\; \texttt{<X>}\ \mathcal{M}(y)\ \texttt{<Y>}\ <S_1>$$

$$<S_1> \;\longrightarrow\; <S_1>\ \texttt{<X>}\ \mathcal{M}(y)\ \texttt{<Y>}$$

$$<S_1>, <S_2> \;\longrightarrow\; <S_1>\ \texttt{<X>}\ \mathcal{M}(y)\ \texttt{<Y>}\ <S_2>$$

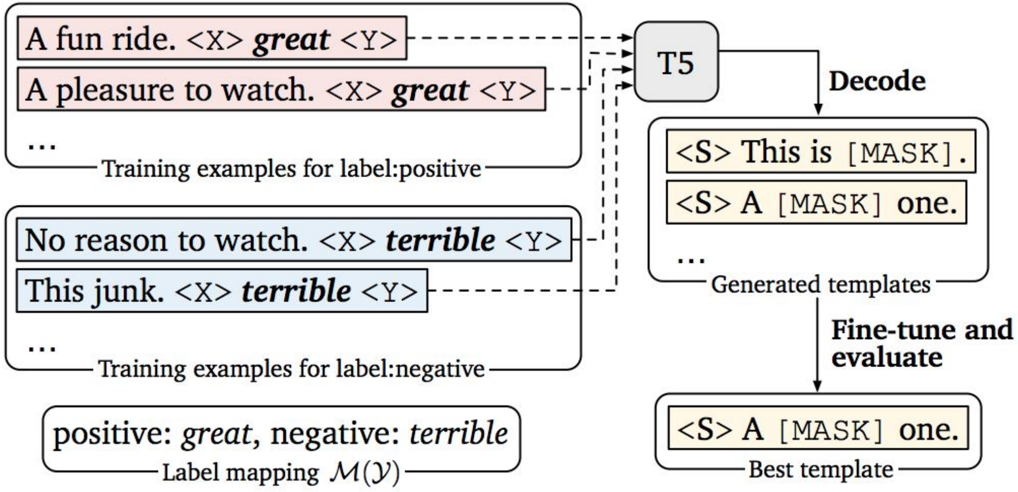


Figure 2: Our approach for template generation.

Choose the best templates

$$\sum_{j=1}^{|\mathcal{T}|} \sum_{(x_{\text{in}},y)\in\mathcal{D}_{\text{train}}} \log P_{\text{T5}}\big(t_j \mid t_1, ..., t_{j-1}, \mathcal{T}_{\text{g}}\big(x_{\text{in}}, y\big)\big), \quad (5)$$

# Finetune with demonstration

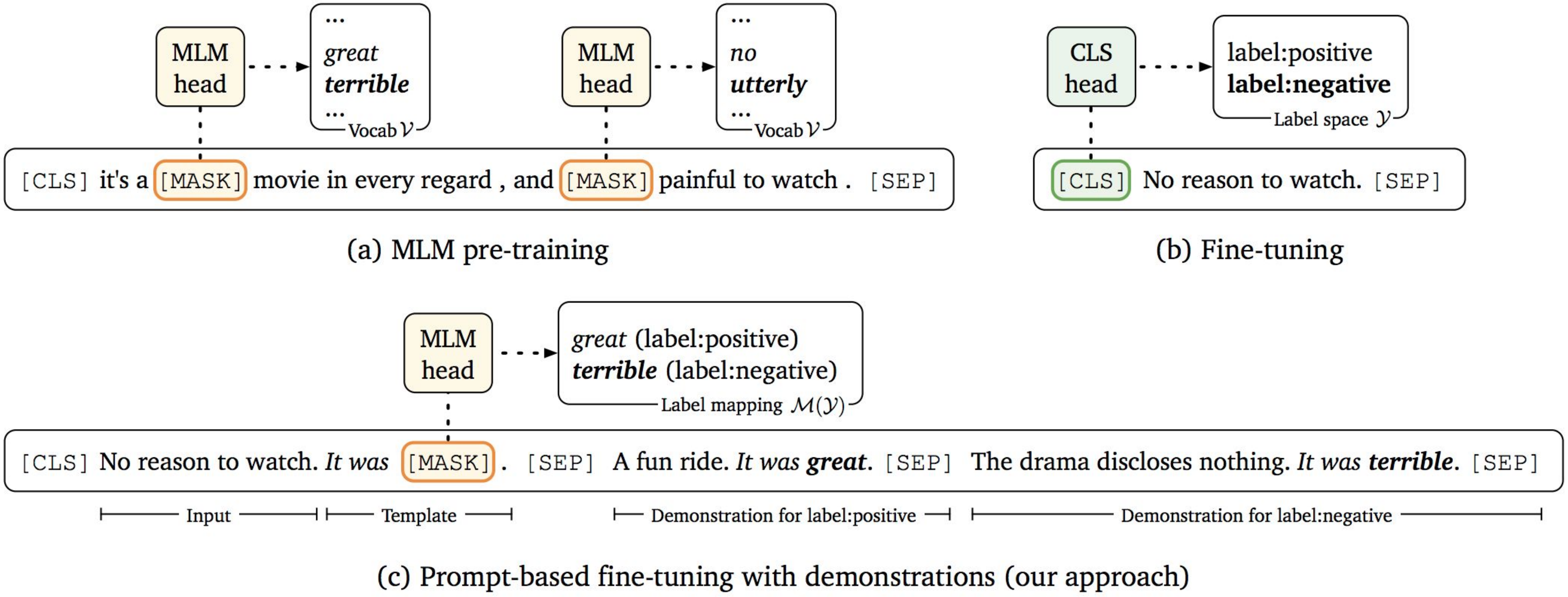


Figure 1: An illustration of (a) masked language model (MLM) pre-training, (b) standard fine-tuning, and (c) our proposed LM-BFF using prompt-based fine-tuning with demonstrations.

# Results

| | **SST-2** (acc) | **SST-5** (acc) | **MR** (acc) | **CR** (acc) | **MPQA** (acc) | **Subj** (acc) | **TREC** (acc) | **CoLA** (Matt.) |
|---|---|---|---|---|---|---|---|---|
| Majority† | *50.9* | *23.1* | *50.0* | *50.0* | *50.0* | *50.0* | *18.8* | *0.0* |
| Prompt-based zero-shot‡ | 83.6 | 35.0 | 80.8 | 79.5 | 67.6 | 51.4 | 32.0 | 2.0 |
| "GPT-3" in-context learning | 84.8 (1.3) | 30.6 (0.9) | 80.5 (1.7) | 87.4 (0.8) | 63.8 (2.1) | 53.6 (1.0) | 26.2 (2.4) | -1.5 (2.4) |
| Fine-tuning | 81.4 (3.8) | 43.9 (2.0) | 76.9 (5.9) | 75.8 (3.2) | 72.0 (3.8) | 90.8 (1.8) | 88.8 (2.1) | **33.9** (14.3) |
| Prompt-based FT (man) | 92.7 (0.9) | 47.4 (2.5) | 87.0 (1.2) | 90.3 (1.0) | 84.7 (2.2) | 91.2 (1.1) | 84.8 (5.1) | 9.3 (7.3) |
| + demonstrations | 92.6 (0.5) | **50.6** (1.4) | 86.6 (2.2) | 90.2 (1.2) | **87.0** (1.1) | **92.3** (0.8) | 87.5 (3.2) | 18.7 (8.8) |
| Prompt-based FT (auto) | 92.3 (1.0) | 49.2 (1.6) | 85.5 (2.8) | 89.0 (1.4) | 85.8 (1.9) | 91.2 (1.1) | 88.2 (2.0) | 14.0 (14.1) |
| + demonstrations | **93.0** (0.6) | 49.5 (1.7) | **87.7** (1.4) | **91.0** (0.9) | 86.5 (2.6) | 91.4 (1.8) | **89.4** (1.7) | 21.8 (15.9) |
| Fine-tuning (full)† | *95.0* | *58.7* | *90.8* | *89.4* | *87.8* | *97.0* | *97.4* | *62.6* |

| | **MNLI** (acc) | **MNLI-mm** (acc) | **SNLI** (acc) | **QNLI** (acc) | **RTE** (acc) | **MRPC** (F1) | **QQP** (F1) | **STS-B** (Pear.) |
|---|---|---|---|---|---|---|---|---|
| Majority† | *32.7* | *33.0* | *33.8* | *49.5* | *52.7* | *81.2* | *0.0* | - |
| Prompt-based zero-shot‡ | 50.8 | 51.7 | 49.5 | 50.8 | 51.3 | 61.9 | 49.7 | -3.2 |
| "GPT-3" in-context learning | 52.0 (0.7) | 53.4 (0.6) | 47.1 (0.6) | 53.8 (0.4) | 60.4 (1.4) | 45.7 (6.0) | 36.1 (5.2) | 14.3 (2.8) |
| Fine-tuning | 45.8 (6.4) | 47.8 (6.8) | 48.4 (4.8) | 60.2 (6.5) | 54.4 (3.9) | 76.6 (2.5) | 60.7 (4.3) | 53.5 (8.5) |
| Prompt-based FT (man) | 68.3 (2.3) | 70.5 (1.9) | 77.2 (3.7) | 64.5 (4.2) | 69.1 (3.6) | 74.5 (5.3) | 65.5 (5.3) | 71.0 (7.0) |
| + demonstrations | **70.7** (1.3) | **72.0** (1.2) | **79.7** (1.5) | **69.2** (1.9) | 68.7 (2.3) | 77.8 (2.0) | **69.8** (1.8) | 73.5 (5.1) |
| Prompt-based FT (auto) | 68.3 (2.5) | 70.1 (2.6) | 77.1 (2.1) | 68.3 (7.4) | **73.9** (2.2) | 76.2 (2.3) | 67.0 (3.0) | 75.0 (3.3) |
| + demonstrations | 70.0 (3.6) | **72.0** (3.1) | 77.5 (3.5) | 68.5 (5.4) | 71.1 (5.3) | **78.1** (3.4) | 67.7 (5.8) | **76.4** (6.2) |
| Fine-tuning (full)† | *89.8* | *89.5* | *92.6* | *93.3* | *80.9* | *91.4* | *81.7* | *91.9* |