AUTOPROMPT: Eliciting Knowledge from Language Models with Automatically Generated Prompts

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, Sameer Singh
EMNLP 2020

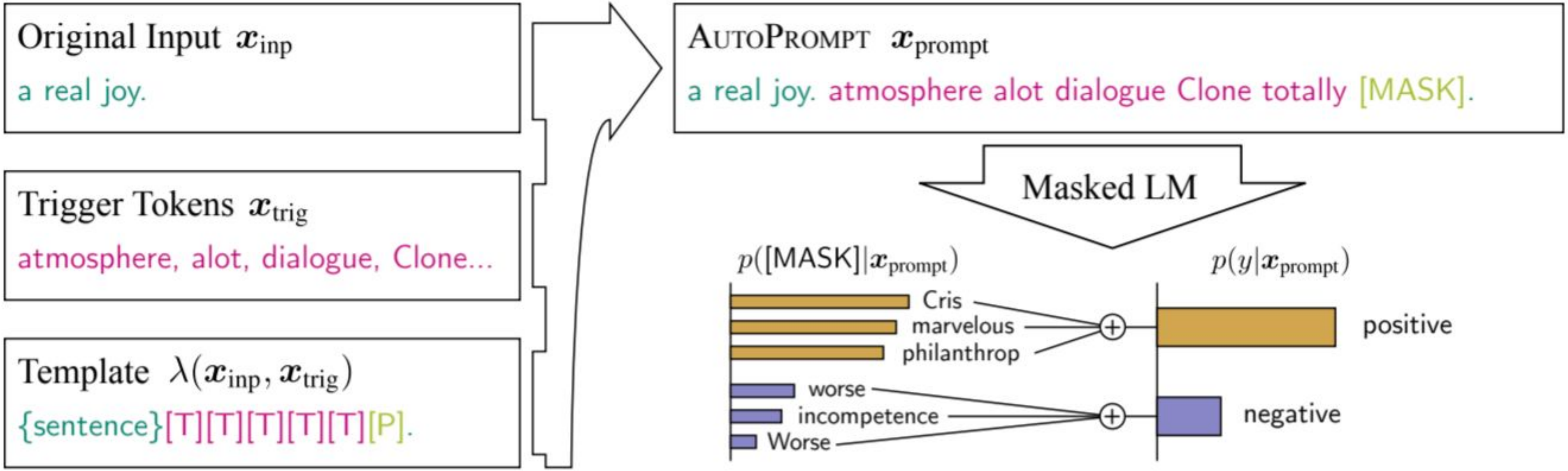
Overview

- 3 ways to explore knowledge in pre-trained LM:
  - Probing -> introduce extra parameters
  - Attention -> might not be interpretable or have spurious correlation
  - Prompting -> It is what the LM is actually doing

- Creating prompts by hand is time-consuming and inefficient

- This paper automatically generates prompts and model different tasks as Masked Language Modeling

Model



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

Model Classification as MLM

1. Input is combined with the trigger words in a pre-defined template

2. The language model predicts $P([MASK]|x_{prompt})$

3. Class probability are computed by marginalizing the probabilities of the label vocabularies:

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

Gradient-Based Prompt Search

1. Trigger tokens are initialized with [MASK]
2. For each trigger token, a candidate set is computed based on the approximated change in the task log-likelihood if the current token is replaced with another one

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

3. For every word in the candidate set the forward pass is repeated and the one with the lowest loss is selected

Automating Label Token Selection

1. In the first step, a feed-forward neural network is trained to make task prediction using the contextualized [MASK] embedding:

$$\boldsymbol{h} = \text{Transformer}_{\text{enc}}(\tilde{\boldsymbol{x}}) \qquad p(y|\boldsymbol{h}^{(i)}) \propto \exp(\boldsymbol{h}^{(i)} \cdot \boldsymbol{y} + \beta_y)$$

2. In the second step, the embedding of the predicted token is fed into the feed-forward neural network:

$$s(y, w) = p(y|\boldsymbol{w}_{\text{out}})$$

3. The vocabulary of the label is selected from high-scoring words:

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

Results

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

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

Thanks