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 $x_{inp}$

```
a real joy.
```

Trigger Tokens $x_{trig}$

```
atmosphere, alot, dialogue, Clone...
```

Template $\lambda(x_{inp}, x_{trig})$

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

**AutoPrompt** $x_{prompt}$

```
a real joy. atmosphere alot dialogue Clone totally [MASK].
```

**Masked LM**

$p([MASK]|x_{prompt})$

```
Cris marvelous philanthrop
```

$p(y|x_{prompt})$

```
positive
```

```
worse
incompetence
Worse
```

```
negative
```
Model Classification as MLM

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

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

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

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

\[ V_{cand} = \text{top-k} \left[ w_{in}^T \nabla \log p(y|x_{\text{prompt}}) \right] \]

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:

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

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

\[ s(y, w) = p(y|w_{\text{out}}) \]

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Dev</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>BiLSTM</td>
<td>-</td>
<td>82.8†</td>
</tr>
<tr>
<td>BiLSTM + ELMo</td>
<td>-</td>
<td>89.3†</td>
</tr>
<tr>
<td>BERT (linear probing)</td>
<td>85.2</td>
<td>83.4</td>
</tr>
<tr>
<td>BERT (finetuned)</td>
<td>-</td>
<td>93.5†</td>
</tr>
<tr>
<td>RoBERTa (linear probing)</td>
<td>87.9</td>
<td>88.8</td>
</tr>
<tr>
<td>RoBERTa (finetuned)</td>
<td>-</td>
<td>96.7†</td>
</tr>
<tr>
<td>BERT (manual)</td>
<td>63.2</td>
<td>63.2</td>
</tr>
<tr>
<td>BERT (AUTO PROMPT)</td>
<td>80.9</td>
<td>82.3</td>
</tr>
<tr>
<td>RoBERTa (manual)</td>
<td>85.3</td>
<td>85.2</td>
</tr>
<tr>
<td>RoBERTa (AUTO PROMPT)</td>
<td>91.2</td>
<td>91.4</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>SICK-E Datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>standard</td>
</tr>
<tr>
<td>Majority</td>
<td>56.7</td>
</tr>
<tr>
<td>BERT (finetuned)</td>
<td>86.7</td>
</tr>
<tr>
<td>BERT (linear probing)</td>
<td>68.0</td>
</tr>
<tr>
<td>RoBERTa (linear probing)</td>
<td>72.6</td>
</tr>
<tr>
<td>BERT (AUTO PROMPT)</td>
<td>62.3</td>
</tr>
<tr>
<td>RoBERTa (AUTO PROMPT)</td>
<td>65.0</td>
</tr>
</tbody>
</table>
Thanks