On a Benefit of Masked Language Modeling: Robustness to Simplicity Bias

Ting-Rui Chiang
tingruic@usc.edu
What we know about pretrained models

- Require less data when fine-tuning
- Smoother loss surface [1]
- Lower intrinsic dimension [2]
- More robust to spurious (unreliable) features [3,4]

[1] Yaru Hao, Li Dong, Furu Wei, and Ke Xu. Visualizing and understanding the effectiveness of BERT
[3] Lifu Tu, Garima Lalwani, Spandana Gella, and He He. An empirical study on robustness to spurious correlations using pre-trained language models.
Why is a model unrobust?

Conjecture: May be due to the **pitfall of simplicity bias** [1].

→ **Simplicity bias:** deep models tend to rely on simple features instead of utilizing all the features [2].

→ **Pitfall:** may not be robust.


Simplicity bias

For example, in the toxic text detection task [1,2]:

The presence (or not) of some group identifiers, e.g. women, black, etc.  
**Single dimension**

The *semantic* encoded by the tokens in the sentence.  
**Much higher dimension**

**Problem: Those spurious features are so tempting!**

---

How can the problem be alleviated?

Data Point $X$

Spurious features: token $X_1$

Robust features: context $X_2$

What if we extract a feature $\Pi = f(X_2)$ such that

1. $\Pi$ is as useful as $X_1$
2. Learning from $\Pi$ is as easy as $X_1$

Effect: Due to the simplicity bias, the model relies more on $\Pi$, and so relies more on $X_2$. 
Theory in this work: MLM extracts $\Pi$

Pretraining phase

Data Point $X$

Spurious features: token $X_1$

Robust features: context $X_2$

Fine-tuning phase

Estimate $P(X_1|X_2)$

$\Pi = P(X_1|X_2)$

Theorem 1: $\Pi$ is as informative as $X_1$ (at least)

Theorem 2: $\Pi$ is as easy as $X_1$ (at least)

Effect: The model relies more on $\Pi$, and so relies more on $X_2$. 
Experimental Settings

- To verify that modeling $P(X_1 | X_2)$ makes models more robust.
- Pretrain two models with two masking policies:
  - Unmask spurious: Remove masks over the spurious features.
  - Unmask random: Remove some masks at random.
- Fine-tune the two models.
- Compare the performance on *out-of-distribution* data.
  - (the spurious features are not useful)
- Two tasks
  - NER: don’t just memorize the name entities.
  - Hate speech detection: don’t rely on the group identifiers.
Results

<table>
<thead>
<tr>
<th>Mask Policy</th>
<th>NER Origin F1 ↑</th>
<th>NER Unseen F1 ↑</th>
<th>Hate Speech Detection All (12893) Accuracy ↑</th>
<th>F1 ↑</th>
<th>Hate Speech Detection NOI (602) Accuracy ↑</th>
<th>FPR ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>scratch vanilla</td>
<td>61.5 0.5</td>
<td>38.7 0.6</td>
<td>83.0 1.6</td>
<td>80.3 1.6</td>
<td>74.8 0.5</td>
<td>46.3 0.3</td>
</tr>
<tr>
<td>unmask random</td>
<td>72.7 0.6</td>
<td>56.5 0.8</td>
<td>83.3 1.1</td>
<td>78.9 1.1</td>
<td>75.8 0.9</td>
<td>25.7 2.3</td>
</tr>
<tr>
<td>unmask spurious</td>
<td>72.9 0.5</td>
<td>53.2 0.8</td>
<td>84.1 0.7</td>
<td>79.8 0.6</td>
<td>73.7 1.0</td>
<td>32.5 2.1</td>
</tr>
<tr>
<td>remove spurious</td>
<td>72.0 0.5</td>
<td>56.7 0.8</td>
<td>84.1 0.7</td>
<td>77.0 0.7</td>
<td>77.3 0.6</td>
<td>21.7 2.0</td>
</tr>
</tbody>
</table>

Modeling the spurious token performs better on OOD.

Similar performance on ID.

Modeling the spurious token indeed improves the robustness.
Conclusion

- Propose the hypothesis why MLM is useful
  - Theoretically: prove that MLM can extract simple features from the robust feature.
  - Empirically: show that modeling the spurious features make models more robust.
Alert: Math Ahead!
My theory: MLM makes models more robust to lexical bias

Data Point X

Simple but spurious features $X_1$

Complex but robust features $X_2$

Assumption 1: from $X$, we can extract $X_1$, $X_2$.

Assumption 2: $X_1$ can predict $Y$ with high accuracy < 100%.

Assumption 3: There is a deterministic mapping from $X_2$ to $Y$. 
**Graphical Model**

Assumption 1: from $X$, we can extract $X_1$, $X_2$.

Assumption 2: $X_1$ can predict $Y$ with high accuracy $< 100\%$.

Assumption 3: There is a deterministic mapping from $X_2$ to $Y$. 
Assumption 1: from $X$, we can extract $X_1$, $X_2$.

Assumption 2: $X_1$ can predict $Y$ with high accuracy $< 100\%$.

Assumption 3: There is a deterministic mapping from $X_2$ to $Y$. 
Graphical Model

Assumption 1: from $X$, we can extract $X_1, X_2$.

Assumption 2: $X_1$ can predict $Y$ with high accuracy $< 100\%$.

Assumption 3: There is a deterministic mapping from $X_2$ to $Y$. 
Lemma

\[ I(X_1; X_2) = I(\Pi; X_1) \]

\[ \Pi := P(X_1 | X_2) \]

\[ P( \mid X_2) \in \text{arg max}_f I(f(X_1); X_1) \]
Theorem 1:

\[ I(\Pi; Y) \geq I(X_1; Y) \]

Lemma 1:

\[ I(X_1; X_2) = I(\Pi; X_1) \]

\(\Pi := P(X_1 | X_2)\)

\(\Pi\) is informative
Theorem 2:

Learning from $\Pi$

- Converges as fast as from $X_1$
- Converges to a solution as good as the optimal solution with $X_1$
- The model is linear

$$\Pi := P(X_1 | X_2)$$
Theorem 2: Formal Results

- Both $\tilde{h}_{X_1}^{(n)}$ and $\tilde{h}_{\Pi}^{(n)}$ converge in $O\left(\frac{1}{n}\right)$
- When $n \to \infty$, the loss of $\tilde{h}_{\Pi}^{(n)}$ is less than $\tilde{h}_{X_1}^{(n)}$.
Theorem 2: Outline of the Proof

- Given \( (x_1^{(1)}, y^{(1)}), (x_1^{(2)}, y^{(2)}), \ldots, (x_1^{(n)}, y^{(n)}) \)

\[
\tilde{h}_{X_1}^{(n)}(Y = 1|X_1 = 1) = \frac{\sum^n_i \mathbb{1}[x_1^{(i)} = 1] \mathbb{1}[y^{(i)} = 1]}{\sum^n_i \mathbb{1}[x_1^{(i)} = 1]}
\]

contains (sort of) underlying dist. of \( x_1 \)

- Given \( (\pi^{(1)}, y^{(1)}), (\pi^{(2)}, y^{(2)}), \ldots, (\pi^{(n)}, y^{(n)}) \)

\[
\tilde{h}_\Pi^{(n)}(Y = 1|X_1 = 1) = \frac{\sum^n_i \pi^{(i)}(X_1 = 1) \mathbb{1}[y^{(i)} = 1]}{\sum^n_i \pi^{(i)}(X_1 = 1)}
\]

\[
\tilde{h}_\Pi^{(n)}(Y = 1|\Pi) = \tilde{h}_\Pi^{(n)}(Y = 1|X_1 = 0)\Pi(X_1 = 0) + \tilde{h}_\Pi^{(n)}(Y = 1|X_1 = 1)\Pi(X_1 = 1)
\]
Alert Lifted