Blank Language Models

Tianxiao Shen*  Victor Quach*  Regina Barzilay  Tommi Jaakkola  (*: equal contribution)
tianxiao@mit.edu
Left-to-Right Language Model

- Generate from scratch

- Start with partially specified text
  - text editing
  - template filling
  - text restoration
  - ...

- They also have ice cream…
Blank Language Model (BLM)

Input: They also have _____ which _____.
Output: They also have ice cream which is really good.

✅ Fine-grained control over generation location
✅ Respect preceding and following context
✅ Variable number of missing tokens
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have _____ which _____. 
Blank Language Model — Overview

• Dynamic canvas where “___” controls where tokens can be placed
• At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
• Stop when there is no “___”

_____ really _____

They also have _____ which ______.
Blank Language Model — Overview

• Dynamic canvas where “___” controls where tokens can be placed.
• At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
• Stop when there is no “___”

They also have _____ which _____ really _____. 
Blank Language Model — Overview

• Dynamic canvas where “___” controls where tokens can be placed
• At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
• Stop when there is no “___”

ice_____

They also have _____ which _____ really ______.
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ice _____ which _____ really _____. 
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word \( w \)
  3. replace that blank with “\( w \)”, “___ \( w \)”, “\( w ___ \)”, or “___ \( w ___ \)”
- Stop when there is no “___”

\[
\text{is} \\
\downarrow \\
\text{They also have ice _____ which _____ really _____.}
\]
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ice _____ which is really ______.
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

```
cream
```

They also have ice _____ which is really ______.
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ice cream which is really good.
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

They also have ice cream which is really good.
Blank Language Model — Overview

- Dynamic canvas where “___” controls where tokens can be placed
- At each step,
  1. select a “___”
  2. predict a word w
  3. replace that blank with “w”, “___ w”, “w ___”, or “___ w ___”
- Stop when there is no “___”

Grammar
- Nonterminal: ___
- Terminals: w ∈ V
- Production rules: ___ → ___? w ___?
  (dist. depends on model and context)
Blank Language Model — Architecture

They also have ____ which ____ .

1) Choose a blank
2) Predict a word
3) Create new blanks

Linear & Softmax

MLP

Fill and repeat
Blank Language Model — Architecture

1) Choose a blank

They also have ____ which ____.
Blank Language Model — Architecture

1) Choose a blank

2) Predict a word

They also have ____ which ____ .

really
Blank Language Model — Architecture

1) Choose a blank

2) Predict a word

3) Create new blanks

They also have ___ which ___ .
Blank Language Model — Architecture

1) Choose a blank

They also have ____ which ____.

2) Predict a word

Linear & Softmax

really

3) Create new blanks

Transformer

MLP

really really really ____

Fill and repeat
# Blank Language Model — Likelihood

A sentence $x$ with $n$ words can be realized by $n!$ trajectories, each corresponds to a different word insertion order

$$p(x; \theta) = \sum_{\sigma \in S_n} p(x, \sigma; \theta) = \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^x, \sigma | c_t^x, \sigma; \theta)$$

<table>
<thead>
<tr>
<th>Step $t$</th>
<th>Canvas $c$</th>
<th>Location $b$</th>
<th>Word $w$</th>
<th>Action $a$</th>
<th>Right blank $r$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td><em>#1</em></td>
<td>#1</td>
<td>have</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>1.</td>
<td><em>#1</em> have <em>#2</em></td>
<td>#1</td>
<td>1</td>
<td>They</td>
<td>No</td>
</tr>
<tr>
<td>2.</td>
<td>They <em>#1</em> have <em>#2</em></td>
<td>#2</td>
<td>10</td>
<td>.</td>
<td>Yes</td>
</tr>
<tr>
<td>3.</td>
<td>They <em>#1</em> have <em>#2</em>.</td>
<td>#2</td>
<td>6</td>
<td>which</td>
<td>Yes</td>
</tr>
<tr>
<td>4.</td>
<td>They <em>#1</em> have <em>#2</em> which <em>#3</em>.</td>
<td>#2</td>
<td>6</td>
<td>which</td>
<td>Yes</td>
</tr>
<tr>
<td>5.</td>
<td>They also have <em>#1</em> which <em>#2</em>.</td>
<td>#2</td>
<td>8</td>
<td>really</td>
<td>Yes</td>
</tr>
<tr>
<td>6.</td>
<td>They also have <em>#1</em> which <em>#2</em> really <em>#3</em>.</td>
<td>#1</td>
<td>4</td>
<td>ice</td>
<td>No</td>
</tr>
<tr>
<td>7.</td>
<td>They also have ice <em>#1</em> which <em>#2</em> really <em>#3</em>.</td>
<td>#2</td>
<td>7</td>
<td>is</td>
<td>No</td>
</tr>
<tr>
<td>8.</td>
<td>They also have ice <em>#1</em> which is really <em>#2</em>.</td>
<td>#1</td>
<td>5</td>
<td>cream</td>
<td>No</td>
</tr>
<tr>
<td>9.</td>
<td>They also have ice cream which is really <em>#1</em>.</td>
<td>#1</td>
<td>9</td>
<td>good</td>
<td>No</td>
</tr>
<tr>
<td>10.</td>
<td>They also have ice cream which is really good.</td>
<td>#1</td>
<td></td>
<td></td>
<td>-End-</td>
</tr>
</tbody>
</table>

**Trajectory**: 1 2 3 4 5 6 7 8 9 10

---

A sentence $x$ with $n$ words can be realized by $n!$ trajectories, each corresponds to a different word insertion order.
Blank Language Model — Training

\[
\log p(x; \theta) = \log \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^x, \sigma | c_t^x, \sigma ; \theta) \quad \text{intractable}
\]

\[
\geq \log(n!) + \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{t=0}^{n-1} \log p(a_t^x, \sigma | c_t^x, \sigma ; \theta)
\]
Blank Language Model — Training

\[
\log p(x; \theta) = \log \sum_{\sigma \in S_n} \prod_{t=0}^{n-1} p(a_t^x, \sigma | c_t^x, \sigma; \theta) \quad \text{intractable}
\]

\[
\log \left( \frac{1}{m} \sum_{i=1}^{m} b_i \right) \geq \frac{1}{m} \sum_{i=1}^{m} \log b_i
\]

\[
\geq \log(n!) + \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{t=0}^{n-1} \log p(a_t^x, \sigma | c_t^x, \sigma; \theta)
\]

1. Uniformly sample \( \sigma \) from \( S_n \)
2. Uniformly sample \( t \) from 0 to \( n - 1 \)
3. Construct canvas \( c_t^x, \sigma \)
4. Compute estimated loss \(- \log(n!) - n \cdot \log p(a_t^x, \sigma | c_t^x, \sigma; \theta)\)
Blank Language Model — Training

$c_t^{x,\sigma}$ only depends on $\sigma_{1:t}$

$\rightarrow$ combine losses of trajectories with the same first $t$ steps and different $(t + 1)$-th step

$$\geq \log(n!) + \frac{1}{n!} \sum_{\sigma \in S_n} \sum_{t=0}^{n-1} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)$$
Blank Language Model — Training

\( c_t^{x, \sigma} \) only depends on \( \sigma_{1:t} \)

→ combine losses of trajectories with the same first \( t \) steps and different \((t + 1)\)-th step

\[
\geq \log(n!) + \sum_{t=0}^{n-1} \frac{1}{n!} \sum_{\sigma \in S_n} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta) \\
= \log(n!) + n \cdot \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \mathbb{E}_{\sigma_{t+1}} \mathbb{E}_{\sigma_{t+2:n}} \left[ \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta) \right] \\
= \log(n!) + \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \left[ \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta) \right]
\]
Blank Language Model — Training

1. Uniformly sample \( t \) from 0 to \( n - 1 \)
2. Uniformly sample \( \sigma_{1:t} \)
3. Construct canvas \( c^x, \sigma_t \)
4. Compute estimated loss
\[
- \log(n!) - \frac{n}{n - t} \sum_{\sigma_{t+1}} \log p(a^x, \sigma | c^x, \sigma; \theta)
\]

\[
= \log(n!) + \mathbb{E}_t \mathbb{E}_{\sigma_{1:t}} \left[ \frac{n}{n - t} \sum_{\sigma_{t+1}} \log p(a^x, \sigma | c^x, \sigma; \theta) \right]
\]

\( n/2 \) action losses per pass :)}
Blank Language Model — Training

1. Uniformly sample $t$ from 0 to $n - 1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c^x,\sigma_t$
4. Compute estimated loss $- \log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c^x,\sigma_t ; \theta)$

$n = 10$

$x = \text{They also have ice cream which is really good .}$
1. Uniformly sample $t$ from 0 to $n - 1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c^x,\sigma_t$
4. Compute estimated loss $-\log(n!) - \frac{n}{n - t} \sum_{\sigma_{t+1}} \log p(a^x,\sigma_t | c^x,\sigma_t ; \theta)$

$n/2$ action losses per pass :)
They also have ice cream which is really good.

$x = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10$

$n = 10$

$t = 5$

$\sigma_{1:t} = (6, 2, 1, 3, 10)$
Blank Language Model — Training

1. Uniformly sample $t$ from 0 to $n - 1$
2. Uniformly sample $\sigma_{1:t}$
3. Construct canvas $c_t^{x,\sigma}$
4. Compute estimated loss

$$- \log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_{t}^{x,\sigma} | c_{t}^{x,\sigma}; \theta)$$

$n/2$ action losses per pass :)

---

$x = \text{They also have ice cream which is really good.}.$

$n = 10$

$t = 5$

$\sigma_{1:t} = (6, 2, 1, 3, 10)$

---

$c_t^{x,\sigma}$ They also have ___ which ___.

---
Blank Language Model — Training

1. Uniformly sample \( t \) from 0 to \( n - 1 \)
2. Uniformly sample \( \sigma_{1:t} \)
3. Construct canvas \( c_t^{x,\sigma} \)
4. Compute estimated loss

\[
- \log(n!) - \frac{n}{n-t} \sum_{\sigma_{t+1}} \log p(a_t^{x,\sigma} | c_t^{x,\sigma}; \theta)
\]

\( x = \) They also have ice cream which is really good.
\( n = 10 \)
\( t = 5 \)
\( \sigma_{1:t} = (6, 2, 1, 3, 10) \)

They also have ice cream which is really good.

n/2 action losses per pass :)
Blank Language Model — Inference

✓ Simple greedy decoding or beam search to fill in the blanks in input
# Experiments — Overview

<table>
<thead>
<tr>
<th>Text Infilling</th>
<th>Input: They also have _____ which _____.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Output: They also have <strong>ice cream</strong> which <strong>is really good</strong>.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ancient Text Restoration</th>
<th>Input: τε εγγονον εισαι????????σοφιαι</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Output: τε εγγονον εισαιου του σοφιαι</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Sentiment Transfer</th>
<th>Input: The employees were <strong>super nice</strong> and <strong>efficient</strong>!</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Output: The employees were <strong>rude</strong> and <strong>unprofessional</strong>!</td>
</tr>
</tbody>
</table>

| Language Modeling   | Output: They also have ice cream which is really good. |
Text Infilling — Dataset

- Yahoo Answers dataset (100K documents, max length 200 words)
- Randomly mask tokens with different ratios
- Contiguous masked tokens → “___”

---

<table>
<thead>
<tr>
<th>Mask Ratio</th>
<th>When time flies, where does it go? to the center of the universe to be recycled and made into new time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>when time flies, _____ does it go? _____ the center of the _____ to be recycled _____ made into new time.</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Mask Ratio</th>
<th>When time flies, where does it go? to the center of the universe to be recycled and made into new time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>50%</td>
<td>when time _____, where _____? _____ the _____ of _____ universe to _____ recycled _____ made into _____.</td>
</tr>
</tbody>
</table>
Text Infilling — Metrics

- Accuracy: BLEU score against original document
- Fluency: perplexity evaluated by a pre-trained left-to-right LM

<table>
<thead>
<tr>
<th>Mask Ratio</th>
<th>when time flies, where does it go? to the center of the universe to be recycled and made into new time.</th>
</tr>
</thead>
<tbody>
<tr>
<td>10%</td>
<td>when time flies, _____ does it go? _____ the center of the _____ to be recycled _____ made into new time.</td>
</tr>
<tr>
<td>50%</td>
<td>when time flies, where does it go? to the center of the universe to be recycled and made into new time.</td>
</tr>
<tr>
<td></td>
<td>when time _____, where _____? _____ the _____ of _____ universe to _____ recycled _____ made into ____.</td>
</tr>
</tbody>
</table>
Text Infilling — Baselines

- Seq2seq-fill [Donahue et al., 2020]
  - output tokens to fill in the blanks, separated by “|”
- Seq2seq-full [Donahue et al., 2020]
  - output full document from input
Text Infilling — Results

- **Seq2seq-fill**
- **Seq2seq-full**

**Failure Rate (%)**

Mask Ratio (%)

- incorrect number of separations
- miss existing tokens
Text Infilling — Baselines

• BERT+LM
  - feed BERT representation of each blank to left-to-right LM that learns to generate tokens in that blank
  - at test time, fill in the blanks one by one
Text Infilling — Baselines

- BERT+LM
- Masked Language Model (MLM) with oracle length
  - replace blanks with the target number of masks
  - fill the masks autoregressively by most-confident-first heuristic
Text Infilling — Baselines

• BERT+LM
• Masked Language Model (MLM) with oracle length
• Insertion Transformer [Stern et al., 2019]
  - cannot specify insertion position
  - force it to generate at valid locations
Text Infilling — Results

![Graph showing BLEU scores for different mask ratios and models]

- **BERT+LM**
- **MLM (oracle length)**
- **InsT**
- **BLM**

The graph displays BLEU scores across different mask ratios (10%, 20%, 30%, 40%, 50%) for the models BERT+LM, MLM (oracle length), InsT, and BLM. The x-axis represents the mask ratio (%), while the y-axis shows the BLEU scores. The graph indicates that BERT+LM generally outperforms the other models across all mask ratios.
Text Infilling — Results

![Bar chart showing perplexity results for different masking ratios and models: BERT+LM, MLM (oracle length), InsT, and BLM. The x-axis represents the mask ratio in percent, ranging from 10% to 50%, and the y-axis represents perplexity. The chart includes a horizontal line indicating the perplexity of the original data.](image-url)
Text Infilling — Examples

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Text Content</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Original</strong></td>
<td>when time flies, <strong>where</strong> does it go? <strong>to</strong> the center of the <strong>universe</strong> to be recycled <strong>and</strong> made into new time.</td>
</tr>
<tr>
<td><strong>Blanked</strong></td>
<td>when time flies, _____ does it go? _____ the center of the _____ to be recycled _____ made into new time.</td>
</tr>
<tr>
<td><strong>BERT+LM</strong></td>
<td>when time flies, <strong>where</strong> does it go? <strong>to</strong> the center of the <strong>earth</strong> to be recycled <strong>came</strong> made into new time.</td>
</tr>
<tr>
<td><strong>MLM (oracle len)</strong></td>
<td>when time flies, <strong>where</strong> does it go? <strong>from</strong> the center of the <strong>earth</strong> to be recycled <strong>converted</strong> made into new time.</td>
</tr>
<tr>
<td><strong>InsT</strong></td>
<td>when time flies, <strong>where</strong> does it go? <strong>for</strong> the center of the <strong>earth</strong> <strong>has</strong> to be recycled <strong>and</strong> made into new time.</td>
</tr>
<tr>
<td><strong>BLM</strong></td>
<td>when time flies, <strong>where</strong> does it go? <strong>for</strong> the center of the <strong>earth</strong> to be recycled <strong>and</strong> made into new time.</td>
</tr>
</tbody>
</table>

Mask Ratio 10%
<table>
<thead>
<tr>
<th></th>
<th>Original</th>
<th>Blanked</th>
<th>BERT+LM</th>
<th>MLM (oracle len)</th>
<th>InsT</th>
<th>BLM</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>when time <em>flies</em>, where <em>does it go?</em> to the center of the universe to be recycled and made into new time.</td>
<td>when time _____, where _____? _____ the _____ of _____ universe to _____ recycled _____ made into _____</td>
<td>when time is , where <em>to?</em> i need to find the way of the universe to be recycled and made into a lot.</td>
<td>when time is , where <em>is the universe?</em> from the creation of the universe to be recycled and made into the universe.</td>
<td>when time was created , where <em>was it?</em> what was the name of the universe to be recycled and made into space.</td>
<td>when time was created , where <em>did it come from?</em> it was the first part of the universe to be recycled and made into space.</td>
</tr>
</tbody>
</table>

Mask Ratio 50%
Ancient Text Restoration — Setup

Ancient Greek Inscriptions dataset (18M characters / 3M words) [Assael et al., 2019]

- number of characters to recover is assumed to be known

Length-aware BLM (L-BLM)

- \([t] \rightarrow [k] w [t-1-k]\)

Baselines [Assael et al., 2019]:

- Pythia: character-level seq2seq model to fill in one slot at a time
- Pythia-word: use both character and word representations
Ancient Text Restoration — Results

Character Error Rate (%)

Human  Pythia  Pythia-word  L-BLM

Mask Ratio (%)

1  25  40  50

Single-slot  Multi-slot
Sentiment Transfer — Approach

1. Remove expressions of high polarity
   - train a sentiment classifier and mask words with attention weight above average
2. Complete the partial sentence with expressions of the target sentiment
   - train two instances of BLM, one for each sentiment
everyone that i spoke with was very helpful and kind.
everyone that i spoke with was rude and unprofessional.
there is definitely not enough room in that part of the venue.
there is always enough parking in that part of the venue.
it is n't terrible, but it is n’t very good either.
it is n’t fancy, but it is still very good either.
Language Modeling — Estimation

Monte-Carlo sampling $p(x; \theta) = \sum_{\sigma \in S_n} p(x, \sigma; \theta) \leftarrow \frac{n!}{m} \sum_{i=1}^{m} p(x, \sigma_i; \theta)$

- estimated perplexity is likely to be higher than actual perplexity
- as $m$ increases, it converges to actual perplexity
Language Modeling — Results

Datasets: Penn Treebank (1M tokens), WikiText-2 (2M), WikiText-103 (103M)

<table>
<thead>
<tr>
<th>Model</th>
<th>PTB</th>
<th>WT2</th>
<th>WT103</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTM (Grave et al., 2016)</td>
<td>82.3</td>
<td>99.3</td>
<td>48.7</td>
</tr>
<tr>
<td>TCN (Bai et al., 2018)</td>
<td>88.7</td>
<td>-</td>
<td>45.2</td>
</tr>
<tr>
<td>AWD-LSTM (Merity et al., 2017)</td>
<td>57.3</td>
<td>65.8</td>
<td>-</td>
</tr>
<tr>
<td>Transformer (Dai et al., 2019)</td>
<td>-</td>
<td>-</td>
<td>30.1</td>
</tr>
<tr>
<td>Adaptive (Baevski and Auli, 2018)</td>
<td>-</td>
<td>-</td>
<td>18.7</td>
</tr>
<tr>
<td>Transformer-XL (Dai et al., 2019)</td>
<td><strong>54.5</strong></td>
<td>-</td>
<td><strong>18.3</strong></td>
</tr>
<tr>
<td>InstT (our implementation)</td>
<td>77.3</td>
<td>91.4</td>
<td>39.4</td>
</tr>
<tr>
<td>BLM</td>
<td>69.2</td>
<td>81.2</td>
<td>42.5</td>
</tr>
</tbody>
</table>

Room for improvements!
Summary

Input: They also have _____ which ______.
Output: They also have ice cream which is really good.

- Dynamically create and fill in blanks
- Effective on text infilling, ancient text restoration, style transfer

More Applications

- Template filling, information fusion, assisting human writing…
- Rewrite to mitigate toxicity and bias
- Representation learning

Extensions

- Add representation for blanks
- Conditional BLM: edit and refine machine translation, dialogue system…