Insertion Transformer: Flexible Sequence Generation via Insertion Operations

Mitchell Stern, William Chan, Jamie Kiros, Jakob Uszkoreit
> Motivation

Approach

Results
Autoregressive models
Autoregressive models

\[ p(\text{word | context}) \]
Autoregressive models

$t=0: \left[ \right]$  \hspace{1cm} p(\text{word} \mid \text{context})
Autoregressive models

\[ t=0: \emptyset \quad \text{p(word | context)} \]
\[ t=1: \text{[three]} \]
Autoregressive models

\[ p(\text{word} \mid \text{context}) \]

\( t=0: [] \)
\( t=1: [\text{three}] \)
\( t=2: [\text{three, friends}] \)
Autoregressive models

\[ p(\text{word} \mid \text{context}) \]

\[
\begin{align*}
  t=0: & \quad [] \\
  t=1: & \quad [\text{three}] \\
  t=2: & \quad [\text{three}, \text{friends}] \\
  t=3: & \quad [\text{three}, \text{friends}, \text{ate}] 
\end{align*}
\]
Autoregressive models

\[ t=0: [] \]
\[ t=1: \text{[three]} \]
\[ t=2: \text{[three, friends]} \]
\[ t=3: \text{[three, friends, ate]} \]
\[ t=4: \text{[three, friends, ate, lunch]} \]
Autoregressive models

\[ \text{p(word | context)} \]

t=0: []

t=1: [three]

t=2: [three, friends]

t=3: [three, friends, ate]

t=4: [three, friends, ate, lunch]

t=5: [three, friends, ate, lunch, together]
Autoregressive models

Examples:

• Recurrent neural networks
• Convolutional neural networks
• Transformers (Vaswani et al., 2017)

→ Accurate but require left-to-right decoding
Non-autoregressive models
Non-autoregressive models

\[ p(\text{all words} \mid \text{context}) \]
Non-autoregressive models

t=0: []  \quad p(\text{all words } | \text{ context})
Non-autoregressive models

t=0: []

\[ p(\text{all words } | \text{ context}) \]

t=1: [three, friends, ate, lunch, together]
Non-autoregressive models

\[ p(\text{all words} \mid \text{context}) \]
Non-autoregressive models

t=0: []

$\text{p(all words | context)}$
Non-autoregressive models

\[ t=0: \quad \] \quad p(\text{all words} \mid \text{context})

\[ t=1: \quad \text{[three, friend, eat, to, lunch]} \]
Non-autoregressive models

\[ t=0: \[] \quad p(\text{all words } | \text{ context}) \]

\[ t=1: [\text{three, friend, eat, to, lunch}] \]

\[ t=2: [\text{three, friends, ate, lunch, together}] \]
Non-autoregressive models

Examples:
• Non-autoregressive Transformer (Gu et al., 2017)
• Iterative refinement Transformer (Lee et al., 2018)

→ Fast but produce lower-quality outputs and require explicit length modeling
A middle ground?
A middle ground?

Desired qualities:

- Flexible generation orders
- Some degree of parallelism
- High-quality outputs
A middle ground?

Desired qualities:

• Flexible generation orders
• Some degree of parallelism
• High-quality outputs

★ Solution: Insertion Transformer!
Motivation

> Approach

Results
Insertion-based models (serial)
Insertion-based models (serial)

\[ p(\text{word}, \text{slot} \mid \text{context}) \]
Insertion-based models (serial)

$t=0$: $[] \quad p(\text{word, slot} \mid \text{context})$
Insertion-based models (serial)

\begin{align*}
  t=0: & \quad [] \\
  t=1: & \quad [\text{ate}] \\
  \quad \quad & \quad p(\text{word, slot} \mid \text{context})
\end{align*}
Insertion-based models (serial)

t=0: []

t=1: [ate]

t=2: [ate, together]
Insertion-based models (serial)

\( t=0: [\] \) \( \quad \) \( p(\text{word, slot | context}) \)
\( t=1: [\text{ate}] \)
\( t=2: [\text{ate, together}] \)
\( t=3: [\text{three, ate, together}] \)
Insertion-based models (serial)

\[
t=0: \left[ \right] \\
t=1: \left[ \text{ate} \right] \\
t=2: \left[ \text{ate, together} \right] \\
t=3: \left[ \text{three, ate, together} \right] \\
t=4: \left[ \text{three, friends, ate, together} \right]
\]

\[p(\text{word, slot} \mid \text{context})\]
Insertion-based models (serial)

\[
\begin{align*}
  t=0: & \quad [] & p(\text{word, slot} \mid \text{context}) \\
  t=1: & \quad [\text{ate}] \\
  t=2: & \quad [\text{ate}, \text{together}] \\
  t=3: & \quad [\text{three}, \text{ate}, \text{together}] \\
  t=4: & \quad [\text{three}, \text{friends}, \text{ate}, \text{together}] \\
  t=5: & \quad [\text{three}, \text{friends}, \text{ate}, \text{lunch}, \text{together}] 
\end{align*}
\]
Insertion-based models (parallel)
Insertion-based models (parallel)

\[ p(\text{word} \mid \text{slot, context}) \]
Insertion-based models (parallel)

t=0: []  \quad \quad p(\text{word} \mid \text{slot, context})
Insertion-based models (parallel)

t=0: []

\[ p(\text{word} \mid \text{slot}, \text{context}) \]

t=1: [ate]
Insertion-based models (parallel)

\[ p(\text{word} \mid \text{slot, context}) \]

\begin{align*}
\text{t=0: } & \, [\, ] \\
\text{t=1: } & \, [\text{ate}] \\
\text{t=2: } & \, [\text{three, ate, together}] \\
\end{align*}
Insertion-based models (parallel)

\[
\begin{align*}
t=0: \ &[] \quad \text{p(word | slot, context)} \\
t=1: \ &\text{[ate]} \\
t=2: \ &\text{[three, ate, together]} \\
t=3: \ &\text{[three, friends, ate, lunch, together]}
\end{align*}
\]
Insertion Transformer
Insertion Transformer

$w_1 \quad w_2 \quad w_3$
<table>
<thead>
<tr>
<th></th>
<th>w₁</th>
<th>w₂</th>
<th>w₃</th>
</tr>
</thead>
<tbody>
<tr>
<td>START</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STOP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Insertion Transformer

Transformer with full self-attention

\[
\langle \text{START} \rangle \quad w_1 \quad w_2 \quad w_3 \quad \langle \text{STOP} \rangle
\]
Insertion Transformer

Transformer with full self-attention

⟨START⟩ h₀ w₁ h₁ w₂ h₂ w₃ ⟨STOP⟩
Insertion Transformer

\[ p(\text{word} \mid \text{slot} = i) \sim \text{softmax} (h_i W_{\text{word}}) \]

\[ p(\text{slot} = i) \sim \text{softmax} (H q_{\text{slot}}) \]
Training
Training

Idea: train model to be able to complete any partial hypothesis
Training

Idea: train model to be able to complete any partial hypothesis

ate   together
Training

Idea: train model to be able to complete any partial hypothesis

\{three, friends\} \quad \text{lunch} \quad \langle\text{EOS}\rangle

\uparrow \quad \text{ate} \quad \uparrow \quad \text{together} \quad \uparrow
Training

Idea: train model to be able to complete any partial hypothesis

\{three, friends\} lunch \langle\text{EOS}\rangle

\uparrow \quad \text{ate} \quad \uparrow \quad \text{together} \quad \uparrow

Training procedure:
Idea: train model to be able to complete any partial hypothesis

{three, friends} lunch ⟨EOS⟩
ate together

Training procedure:
1. Sample random subset size $k \sim \text{Uniform}([0, 1, ..., n])$
Training

Idea: train model to be able to complete any partial hypothesis

\{three, friends\} lunch \langle\text{EOS}\rangle

\uparrow \text{ate} \uparrow \text{together} \uparrow

Training procedure:
1. Sample random subset size \( k \sim \text{Uniform}([0, 1, \ldots, n]) \)
2. Sample random subset of \( k \) tokens to obtain partial output
Training

Idea: train model to be able to complete any partial hypothesis

\{three, friends\} lunch \langle\text{EOS}\rangle

\uparrow \text{ate} \uparrow \text{together} \uparrow

Training procedure:
1. Sample random subset size \(k \sim \text{Uniform}([0, 1, \ldots, n])\)
2. Sample random subset of \(k\) tokens to obtain partial output
3. Compute loss for single insertion step
Uniform loss
Uniform loss

Give uniform weight to correct words within each slot:
Uniform loss

Give uniform weight to correct words within each slot:

\[
\text{slot-loss}(x, \hat{y}, l) = \frac{1}{j_l - i_l + 1} \sum_{i=i_l}^{j_l} - \log p(y_i, l | x, \hat{y})
\]
Uniform loss

Give uniform weight to correct words within each slot:

\[
\text{slot-loss}(x, \hat{y}, l) = \frac{1}{j_l - i_l + 1} \sum_{i=i_l}^{j_l} - \log p(y_i, l \mid x, \hat{y})
\]

\[
\text{loss}(x, \hat{y}) = \frac{1}{k + 1} \sum_{l=0}^{k} \text{slot-loss}(x, \hat{y}, l)
\]
Balanced binary tree loss
Balanced binary tree loss

Give higher weight to words closer to the center of each slot:
Balanced binary tree loss

Give higher weight to words closer to the center of each slot:

This encourages a balanced binary tree generation order, allowing for maximal parallelism during generation.
Balanced binary tree loss
Balanced binary tree loss

\[
\text{slot-loss}(x, \hat{y}, l) = \sum_{i=i_l}^{j_l} - \log p(y_i, l \mid x, \hat{y}) \cdot w_l(i)
\]
Balanced binary tree loss

\[
\text{slot-loss}(x, \hat{y}, l) = \sum_{i = i_l}^{j_l} - \log p(y_i, l \mid x, \hat{y}) \cdot w_l(i)
\]

\[
\text{loss}(x, \hat{y}) = \frac{1}{k + 1} \sum_{l=0}^{k} \text{slot-loss}(x, \hat{y}, l)
\]
Balanced binary tree loss

\[
\text{slot-loss}(x, \hat{y}, l) = \sum_{i=i_l}^{j_l} - \log p(y_i, l \mid x, \hat{y}) \cdot w_l(i)
\]

\[
\text{loss}(x, \hat{y}) = \frac{1}{k + 1} \sum_{l=0}^{k} \text{slot-loss}(x, \hat{y}, l)
\]

where

\[
w_l(i) = \frac{\exp(-d_l(i)/\tau)}{\sum_{i' = i_l}^{j_l} \exp(-d_l(i')/\tau)}
\]

\[
d_l(i) = \left| \frac{i_l + j_l}{2} - i \right|
\]
Inference
Inference

Greedy decoding: pick best \((content, location)\) pair

\[
(\hat{c}_t, \hat{l}_t) = \arg\max_{c,l} p(c, l \mid x, \hat{y}_t)
\]
Inference

Greedy decoding: pick best (content, location) pair

$$(\hat{c}_t, \hat{l}_t) = \underset{c,l}{\text{argmax}} p(c,l \mid x, \hat{y}_t)$$

Parallel decoding: pick best content for each location

$$\hat{c}_{l,t} = \underset{c}{\text{argmax}} p(c \mid l, x, \hat{y}_t)$$
Inference

Greedy decoding: pick best (content, location) pair

\[(\hat{c}_t, \hat{l}_t) = \arg\max_{c,l} p(c, l \mid x, \hat{y}_t)\]

Parallel decoding: pick best content for each location

\[\hat{c}_{l,t} = \arg\max_{c} p(c \mid l, x, \hat{y}_t)\]

Slots emit \langle EOS\rangle token to stop
Motivation

Approach

> Results
Examples

Input: Everyone has the Internet, an iPad and eBooks.

Output: Jeder hat das Internet, ein iPad und eBooks.

Greedy decode (uniform loss):

Jeder hat das Internet, ein iPad und eBooks.

(continued)

Jeder hat das Internet, ein iPad und eBooks.
Examples

**Input:** They want to create a post on the college’s equal opportunities committee to ensure that their opinions can be aired freely.

**Output:** Sie wollen einen Posten im Ausschuss für Chancengleichheit des Kollegiums einrichten, um sicherzustellen, dass ihre Meinungen frei zur Sprache gebracht werden können.

**Parallel decode (uniform loss):**

Sie wollen einen Posten im Ausschuss für Chancengleichheit des Koll egi ums. einrichten, um sicherzustellen, dass ihre Mein ungen. frei zur Sprache gebracht werden können...
Examples

**Input:** But on the other side of the state, that is not the impression many people have of their former governor.

**Output:** Aber auf der anderen Seite des Staates ist das nicht der Eindruck, den viele von ihrem ehemaligen Gouverneur haben.

**Parallel decode (binary tree loss):**

Aber auf der anderen Seite des Staates ist das nicht der Eindruck, den viele von ihrem ehemaligen Gouverneur haben.
# Results (WMT 2014 En-De Test)

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive Left-to-Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformer (Vaswani et al., 2017)</td>
<td>27.3</td>
<td>$n$</td>
</tr>
<tr>
<td>Semi-Autoregressive Left-to-Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAT (Wang et al., 2018)</td>
<td>24.83</td>
<td>$n/6$</td>
</tr>
<tr>
<td>Blockwise Parallel (Stern et al., 2018)</td>
<td>27.40</td>
<td>$\approx n/5$</td>
</tr>
<tr>
<td>Non-Autoregressive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAT (Gu et al., 2018)</td>
<td>17.69</td>
<td>1</td>
</tr>
<tr>
<td>Iterative Refinement (Lee et al., 2018)</td>
<td>21.61</td>
<td>10</td>
</tr>
<tr>
<td>Our Approach (Greedy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insertion Transformer + Left-to-Right</td>
<td>23.94</td>
<td>$n$</td>
</tr>
<tr>
<td>Insertion Transformer + Binary Tree</td>
<td>27.29</td>
<td>$n$</td>
</tr>
<tr>
<td>Insertion Transformer + Uniform</td>
<td>27.12</td>
<td>$n$</td>
</tr>
<tr>
<td>Our Approach (Parallel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insertion Transformer + Binary Tree</td>
<td>27.41</td>
<td>$\approx \log_2 n$</td>
</tr>
<tr>
<td>Insertion Transformer + Uniform</td>
<td>26.72</td>
<td>$\approx \log_2 n$</td>
</tr>
</tbody>
</table>
## Results (WMT 2014 En-De Test)

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th>Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Autoregressive Left-to-Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transformer (Vaswani et al., 2017)</td>
<td>27.3</td>
<td>$n$</td>
</tr>
<tr>
<td>Semi-Autoregressive Left-to-Right</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SAT (Wang et al., 2018)</td>
<td>24.83</td>
<td>$n/6$</td>
</tr>
<tr>
<td>Blockwise Parallel (Stern et al., 2018)</td>
<td>27.40</td>
<td>$\approx n/5$</td>
</tr>
<tr>
<td>Non-Autoregressive</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NAT (Gu et al., 2018)</td>
<td>17.69</td>
<td>1</td>
</tr>
<tr>
<td>Iterative Refinement (Lee et al., 2018)</td>
<td>21.61</td>
<td>10</td>
</tr>
<tr>
<td>Our Approach (Greedy)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insertion Transformer + Left-to-Right</td>
<td>23.94</td>
<td>$n$</td>
</tr>
<tr>
<td>Insertion Transformer + Binary Tree</td>
<td>27.29</td>
<td>$n$</td>
</tr>
<tr>
<td>Insertion Transformer + Uniform</td>
<td>27.12</td>
<td>$n$</td>
</tr>
<tr>
<td>Our Approach (Parallel)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insertion Transformer + Binary Tree</td>
<td>27.41</td>
<td>$\approx \log_2 n$</td>
</tr>
<tr>
<td>Insertion Transformer + Uniform</td>
<td>26.72</td>
<td>$\approx \log_2 n$</td>
</tr>
</tbody>
</table>
Parallel decoding analysis
Parallel decoding analysis

Iteration reduction vs. autoregressive models:
Parallel decoding analysis

Iteration reduction vs. autoregressive models:

Wall-clock speedup vs. T2T Transformer:
- 4.19x averaged over dev set
- 7.28x on longest 10% of sentences
Thanks!

Pacific Ballroom Poster #46
Extra Slides
### Results (dev set)

<table>
<thead>
<tr>
<th>Loss</th>
<th>Termination</th>
<th>BLEU (+EOS)</th>
<th>BLEU (+EOS)</th>
<th>BLEU (+EOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>+Distillation</td>
<td>+Distillation, +Parallel</td>
<td></td>
</tr>
<tr>
<td>Left-to-Right</td>
<td>Sequence</td>
<td>20.92 (20.92)</td>
<td>23.29 (23.36)</td>
<td>-</td>
</tr>
<tr>
<td>Binary Tree ($\tau = 0.5$)</td>
<td>Slot</td>
<td>20.35 (21.39)</td>
<td>24.49 (25.55)</td>
<td>25.33 (25.70)</td>
</tr>
<tr>
<td>Binary Tree ($\tau = 1.0$)</td>
<td>Slot</td>
<td>21.02 (22.37)</td>
<td>24.36 (25.43)</td>
<td>25.43 (25.76)</td>
</tr>
<tr>
<td>Binary Tree ($\tau = 2.0$)</td>
<td>Slot</td>
<td>20.52 (21.95)</td>
<td>24.59 (25.80)</td>
<td>25.33 (25.80)</td>
</tr>
<tr>
<td>Uniform</td>
<td>Sequence</td>
<td>19.34 (22.64)</td>
<td>22.75 (25.45)</td>
<td>-</td>
</tr>
<tr>
<td>Uniform</td>
<td>Slot</td>
<td>18.26 (22.16)</td>
<td>22.39 (25.58)</td>
<td>24.31 (24.91)</td>
</tr>
</tbody>
</table>
Results (dev set, model variants)

<table>
<thead>
<tr>
<th>Joint</th>
<th>Contextual</th>
<th>Mixture</th>
<th>BLEU (+EOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>✗</td>
<td>✗</td>
<td>✗</td>
<td>22.39 (25.58)</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>✗</td>
<td>22.92 (25.14)</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>✗</td>
<td>23.00 (25.41)</td>
</tr>
<tr>
<td>✗</td>
<td>✗</td>
<td>✓</td>
<td>22.19 (25.58)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✗</td>
<td>23.22 (25.44)</td>
</tr>
<tr>
<td>✓</td>
<td>✗</td>
<td>✓</td>
<td>20.17 (24.19)</td>
</tr>
<tr>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>23.29 (25.48)</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>22.16 (25.44)</td>
</tr>
</tbody>
</table>
Results (dev set, parallel decoding)

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU (+EOS)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Binary Tree ($\tau = 0.5$)</td>
<td>25.33 (25.70)</td>
</tr>
<tr>
<td>Binary Tree ($\tau = 1.0$)</td>
<td>25.43 (25.76)</td>
</tr>
<tr>
<td>Binary Tree ($\tau = 2.0$)</td>
<td>25.33 (25.80)</td>
</tr>
<tr>
<td>Uniform</td>
<td>24.31 (24.91)</td>
</tr>
<tr>
<td>Uniform + Contextual</td>
<td>24.54 (24.74)</td>
</tr>
<tr>
<td>Uniform + Mixture</td>
<td>24.33 (25.11)</td>
</tr>
<tr>
<td>Uniform + Contextual + Mixture</td>
<td>24.68 (25.02)</td>
</tr>
</tbody>
</table>