XLNet: Generalized Autoregressive Pre-training for Language Understanding

Zhilin Yang*, Zihang Dai*, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, Quoc V. Le

Presented by Andrew Or, Ksenia Sokolova
Outline

**XLNet Key Ideas:** high-level comparison with BERT

**XLNet Backbone:** Transformer-XL

**Pre-training Objectives:** comparison with AR and BERT

**XLNet Design:** permutation, masks, two-stream attention

**Results:** XLNet outperforms BERT on 20 tasks
Background

★ Before: autoregressive (ex. ELMo, GPT) and autoencoding (ex. BERT) models are the two most successful pre-training objectives

★ Both approaches have their own limitations
Autoregressive Models

Use context to predict the next word

\[
p(x) = \prod_{t=1}^{T} p(x_t \mid x_{<t})
\]

\[
p(x) = \prod_{t=T}^{1} p(x_t \mid x_{>t})
\]

Only considers context in one direction
Autoencoding Models (BERT)

Note: previously a SOTA pretraining approach

Fine-tuning discrepancy caused by [MASK] tokens (not in real data)

Peter has a [MASK] that does not like [MASK]

Assumes *cat* and *yarn* are independent, which is wrong

No joint probability between masked entries
Two Notable Objectives for Language Pretraining

Auto-regressive Language Modeling

- York is a city [EOS]
- New York is a city

Unidirectional Transformer

- Full Auto-regressive Dependence
- Free from artificial Noise
- No Bidirectional Context

Denoising Auto-encoding (BERT)

- York is
- New [MASK][MASK] a city

Bidirectional Transformer

- Independent Predictions
- Artificial Noise: [MASK]
- Natural Bidirectional Context
XLNet Key Ideas

*Autoregressive*: use context to predict the next word

*Bidirectional context* from permutation language modeling

*Self-attention* mechanisms, uses Transformer-XL backbone
XLNet Key Ideas

*Autoregressive*: use context to predict the next word

*Bidirectional context* from permutation language modeling

Peter’s **cat** likes yarn

* Peter’s cat likes yarn  yarn Peter’s cat likes
* Peter’s cat yarn likes  yarn Peter’s likes cat
* Peter’s likes cat yarn  yarn cat Peter’s likes
* Peter’s likes yarn cat  yarn cat likes Peter
* Peter’s yarn cat likes  yarn likes Peter’s cat
* Peter’s yarn likes cat  yarn likes cat Peter’s
XLNet Key Ideas

*Autoregressive*: use context to predict the next word

*Bidirectional context* from permutation language modeling

*Peter’s cat likes yarn*

<table>
<thead>
<tr>
<th>Peter’s cat likes yarn</th>
<th>yarn Peter’s cat likes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peter’s cat yarn likes</td>
<td>yarn Peter’s likes cat</td>
</tr>
<tr>
<td>Peter’s likes cat yarn</td>
<td>yarn cat Peter’s likes</td>
</tr>
<tr>
<td>Peter’s likes yarn cat</td>
<td>yarn cat likes Peter</td>
</tr>
<tr>
<td>Peter’s yarn cat likes</td>
<td>yarn likes Peter’s cat</td>
</tr>
<tr>
<td>Peter’s yarn likes cat</td>
<td>yarn likes cat Peter’s</td>
</tr>
</tbody>
</table>
XLNet Key Ideas

*Autoregressive*: use context to predict the next word

*Bidirectional context* from permutation language modeling

*Self-attention* mechanisms, uses Transformer-XL backbone
Transformer-XL

Increases context through segment-level recurrence and a novel positional encoding scheme
Transformer-XL

Increases context through **segment-level recurrence** and a novel positional encoding scheme

- **Cache and reuse hidden state** from the previous segment
- Allows **variable-length context**, great for capturing long-term dependencies
- Resolves the problem of context fragmentation
Before (no segment-level recurrence)
After segment-level recurrence
Transformer-XL

Increases context through segment-level recurrence and a novel positional encoding scheme

Need a way to keep positional information coherent when we reuse the states

- In the original Transformer: *absolute* position within a segment is used
- Need to encode *relative* position

![Original Transformer Diagram](image)
Before
After
Training objectives

Traditional AR models vs BERT vs XLNet
Traditional AR models

\[
\max_{\theta} \log p_\theta(x) = \sum_{t=1}^{T} \log p_\theta(x_t | x_{<t}) = \sum_{t=1}^{T} \log \frac{\exp \left( h_\theta(x_{1:t-1})^T e(x_t) \right)}{\sum_{x'} \exp \left( h_\theta(x_{1:t-1})^T e(x') \right)}
\]

\[
\max_{\theta} \log p_\theta(\hat{x} | \hat{x}) \approx \sum_{t=1}^{T} m_t \log p_\theta(x_t | \hat{x}) = \sum_{t=1}^{T} m_t \log \frac{\exp \left( H_\theta(\hat{x})_t^T e(x_t) \right)}{\sum_{x'} \exp \left( H_\theta(\hat{x})_t^T e(x') \right)}
\]

= 1 if masked

BERT

\[
\max_{\theta} \mathbb{E}_{z \sim Z_T} \left[ \sum_{t=1}^{T} \log p_\theta(x_{z_t} | x_{z_{<t}}) \right]
\]

Set of all permutations

XLMNet*
XLNet Design

Permutation only on factorization order, not the original sequence order

Attention masks provide the context for each prediction

Two-stream self-attention allows prediction to be aware of target position

Partial prediction: only predict $1/K$ tokens in each permutation
**XLNet Design**

**Permutation** only on *factorization order*, not the original *sequence order*

Attention masks provide the context for each prediction

Two-stream self-attention allows prediction to be aware of target position

Partial prediction: only predict $1/K$ tokens in each permutation
Context Depends on the **Factorization Order**

- **Standard LM:** Left-to-right factorization $1 \rightarrow 2 \rightarrow 3 \rightarrow 4$

$$P(x) = P(x_1)P(x_2 | x_1)P(x_3 | x_{1,2})P(x_4 | x_{1,2,3}) \cdots$$
Context Depends on the **Factorization Order**

- Change the Factorization order to: \(4 \rightarrow 1 \rightarrow 3 \rightarrow 2\)

\[ P(\mathbf{x}) = P(x_4)P(x_1 \mid x_4)P(x_3 \mid x_{1,4})P(x_2 \mid x_{1,2,4}) \cdots \]

![Diagram showing the change in factorization order from left to right, with arrows indicating the new order: \(x_1 \leftarrow x_3 \rightarrow x_4\).]
Permutation Language Modeling

- Given a sequence $\mathbf{x}$ of length $T$
- Uniformly sample a factorization order $\mathbf{z}$ from all possible permutations
- Maximize the permutated log-likelihood

$$
\mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[ \log P(\mathbf{x} \mid \mathbf{z}) \right] = \mathbb{E}_{\mathbf{z} \sim \mathcal{Z}_T} \left[ \sum_{t=1}^{T} P(x_{z_t} \mid x_{z_{<t}}, z_t) \right]
$$
XLNet Design

Permutation only on factorization order, not the original sequence order

Attention masks provide the context for each prediction

Two-stream self-attention allows prediction to be aware of target position

Partial prediction: only predict $1/K$ tokens in each permutation
Attention Masks Provide Context

\[
\begin{align*}
\text{Attention output} &= w_1 \cdot v_1 + w_2 \cdot v_2 + w_3 \cdot v_3 + w_4 \cdot v_4 \\
\end{align*}
\]

\[
\begin{align*}
w_1 &= f(q_3 \cdot k_1) \\
w_2 &= f(q_3 \cdot k_2) \\
w_3 &= f(q_3 \cdot k_3) \\
w_4 &= f(q_3 \cdot k_4) \\
f &= \text{softmax + scale}
\end{align*}
\]
Attention Masks Provide Context

\[ w_1 = f(q_3 \cdot k_1) \]

\[ w_2 = 0 \]

\[ w_3 = 0 \]

\[ w_4 = f(q_3 \cdot k_4) \]

\[ f = \text{softmax} + \text{scale} \]

Attention output = \( w_1 \cdot v_1 + w_4 \cdot v_4 \)
XLNet Design

Permutation only on factorization order, not the original sequence order.

Attention masks provide the context for each prediction.

Two-stream self-attention allows prediction to be aware of target position.

Partial prediction: only predict \(1/K\) tokens in each permutation.
Standard AR Parameterization Fails

The apple was eaten

\[ z_1 = \text{apple was } \text{the} \text{ eaten} \quad \text{p}(x | \text{apple, was}) \]

\[ z_2 = \text{apple was } \text{eaten} \text{ the} \quad \text{p}(x | \text{apple, was}) \]

Predicting the and eaten uses the same distribution

Fails to take into account target position
Standard AR Parameterization Fails

The apple was eaten

\[ z_1 = \text{apple was the eaten} \quad p(x | \text{apple, was, [pos=0]}) \]
\[ z_2 = \text{apple was eaten the} \quad p(x | \text{apple, was, [pos=3]}) \]

Predicting *the* and *eaten* uses the same distribution

Fails to take into account **target position**
Reparameterization

- Standard Softmax does **NOT** work

\[
P(x_{zt} \mid x_{z_{<t}}, z_t) = \frac{\exp (e(x_{zt})^\top h(x_{z_{<t}}))}{\sum_{x'} \exp (e(x')^\top h(x_{z_{<t}}))}
\]

- **Proposed** solution: incorporate \(z_t\) into **hidden states**

\[
P(x_{zt} \mid x_{z_{<t}}, z_t) = \frac{\exp (e(x_{zt})^\top g(z_t, x_{z_{<t}}))}{\sum_{x'} \exp (e(x')^\top g(z_t, x_{z_{<t}}))}
\]

Implement this using **two-stream** architecture
Target Position Aware Representation: \( g(z_t, x_{z<t}) \)

Reuse the Idea of Attention

- Stand at the target position \( z_t \)
- Gather information from \( x_{z<t} \)

\[
g(z_t, x_{z<t}) = \text{Attn}_\theta \left( \begin{array}{c}
Q = \text{Enc}(z_t), \\
KV = h(x_{z<t})
\end{array} \right)
\]

Old View

New View
Contradiction: Predicting Self and Others

- Factorization order: $4 \rightarrow 1 \rightarrow 3 \rightarrow 2$

Use $g_1^{(1)}$ to predict $x_1$ (self)

Use $g_1^{(1)}$ to predict $x_3$ (other)

Should not encode $x_1$

Should encode $x_1$
Two-Stream Self-Attention

\[ g_{zt}^{(m)} \leftarrow \text{Attention}(Q = g_{zt}^{(m-1)}, KV = h_{z<zt}^{(m-1)}; \theta), \quad \text{(query stream: use } z_t \text{ but cannot see } x_{z_t}) \]

\[ h_{zt}^{(m)} \leftarrow \text{Attention}(Q = h_{zt}^{(m-1)}, KV = h_{z\leq t}^{(m-1)}; \theta), \quad \text{(content stream: use both } z_t \text{ and } x_{z_t}) \]

**Query stream** encodes target position information \((z_t)\)

**Content stream** encodes both context and the target word \((x_{z_t})\)

\[
p_{\theta}(X_{zt} = x \mid x_{z<zt}) = \frac{\exp((e(x)^\top g_{\theta}(x_{z<zt}, z_t))}{\sum_{x'} \exp((e(x')^\top g_{\theta}(x_{z<zt}, z_t)))}
\]
Two-Stream Self-Attention

\[ g_{z_t}^{(m)} \leftarrow \text{Attention}(Q = g_{z_t}^{(m-1)}, KV = h_{z_t}^{(m-1)}; \theta), \] (query stream: use \( z_t \) but cannot see \( x_{z_t} \))

\[ h_{z_t}^{(m)} \leftarrow \text{Attention}(Q = h_{z_t}^{(m-1)}, KV = h_{z_t}^{(m-1)}; \theta), \] (content stream: use both \( z_t \) and \( x_{z_t} \)).
Two-Stream Attention

- Factorization order: $4 \rightarrow 1 \rightarrow 3 \rightarrow 2$

**Encoding.** Predicting $x_2$ and $x_3$ (others).

**Decoding.** Predicting $x_1$ (self).

$h_1$ encodes $x_1$

$g_1$ does not encode $x_1$
Masked Two-stream Attention

Attention Masks

Content stream: can see self

Query stream: cannot see self

Sample a factorization order: 3 → 2 → 4 → 1
**XLNet Design**

**Permutation** only on *factorization order*, not the original *sequence order*.

**Attention masks** provide the context for each prediction.

**Two-stream self-attention** allows prediction to be aware of target position.

**Partial prediction:** only predict $1/K$ tokens in each permutation.
Partial Prediction

Motivation: reduce optimization difficulty from too little context

Split sequence into **context words** and **target words**, cut off at $c$

Only predict target words ($1/K$ of original sequence)

\[
\max_{\theta} \mathbb{E}_{z \sim z_T} \left[ \log p_{\theta}(x_{z > c} \mid x_{z \leq c}) \right] = \mathbb{E}_{z \sim z_T} \left[ \sum_{t=c+1}^{\mid z \mid} \log p_{\theta}(x_{zt} \mid x_{z < t}) \right]
\]

\[
\left\lfloor \frac{\mid z \mid}{\mid z \mid - c} \right\rfloor \approx K = 6 \quad (\sim 17\% \text{ target})
\]

XLNet-Large
Example: Comparison with BERT

Input sentence: New York is a city, masked *New* and *York*

XLNet factorization order: [is, a, city, New, York]

\[
\log p(\text{New York} \mid \text{is a city})
\]

\[
J_{\text{BERT}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{is a city}),
\]

\[
J_{\text{XLNet}} = \log p(\text{New} \mid \text{is a city}) + \log p(\text{York} \mid \text{New}, \text{is a city}).
\]
Evaluation

Comparison with BERT
Experiment 1: Comparison with BERT

- **Same training data as in BERT**: Wikipedia + BooksCorpus

- **Same hyperparameters for pretraining as in BERT**
  - Model size: L=24, H=1024, A=16
  - Batch size: 256
  - Number of steps: 1M
  - ...

- **Same hyperparameter search space for finetuning as in BERT**
XLNet outperforms BERT on 20 tasks

We report the best of 3 BERT variants. Almost identical training recipes.
Experiment 2: Comparison with RoBERTa

- Less training data for XLNet: 126GB vs 160GB
- Same hyperparameters for pretraining as in RoBERTa
  - Model size: $L=24$, $H=1024$, $A=16$
  - Batch size: 8192
  - Number of steps: 500K
  - ...
- Same hyperparameter search space for finetuning as in RoBERTa
<table>
<thead>
<tr>
<th></th>
<th>MNLI</th>
<th>QNLI</th>
<th>QQP</th>
<th>RTE</th>
<th>SST</th>
<th>MRPC</th>
<th>CoLA</th>
<th>STS</th>
<th>WNLI</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>XLNet&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>89.8/-</td>
<td>93.9</td>
<td>91.8</td>
<td>83.8</td>
<td>95.6</td>
<td>89.2</td>
<td>63.6</td>
<td>91.8</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>RoBERTa</td>
<td><strong>90.2/90.2</strong></td>
<td><strong>94.7</strong></td>
<td><strong>92.2</strong></td>
<td><strong>86.6</strong></td>
<td><strong>96.4</strong></td>
<td><strong>90.9</strong></td>
<td><strong>68.0</strong></td>
<td><strong>92.4</strong></td>
<td><strong>91.3</strong></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ensembles on test (from leaderboard as of July 25, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALICE</td>
<td>88.2/87.9</td>
<td>95.7</td>
<td><strong>90.7</strong></td>
<td>83.5</td>
<td>95.2</td>
<td>92.6</td>
<td><strong>68.6</strong></td>
<td>91.1</td>
<td>80.8</td>
<td>86.3</td>
</tr>
<tr>
<td>MT-DNN</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
<td>87.6</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.2/89.8</td>
<td>98.6</td>
<td>90.3</td>
<td>86.3</td>
<td><strong>96.8</strong></td>
<td><strong>93.0</strong></td>
<td>67.8</td>
<td>91.6</td>
<td><strong>90.4</strong></td>
<td>88.4</td>
</tr>
<tr>
<td>RoBERTa</td>
<td><strong>90.8/90.2</strong></td>
<td><strong>98.9</strong></td>
<td>90.2</td>
<td><strong>88.2</strong></td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td><strong>92.2</strong></td>
<td>89.0</td>
<td><strong>88.5</strong></td>
</tr>
<tr>
<td>Model</td>
<td>MNLI</td>
<td>QNLI</td>
<td>QQP</td>
<td>RTE</td>
<td>SST</td>
<td>MRPC</td>
<td>CoLA</td>
<td>STS</td>
<td>WNLI</td>
<td>Avg</td>
</tr>
<tr>
<td>--------------</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
<td>-----</td>
</tr>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT\textsubscript{LARGE}</td>
<td>86.6/</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XLNet\textsubscript{LARGE}</td>
<td>89.8/</td>
<td>93.9</td>
<td>91.8</td>
<td>83.8</td>
<td>95.6</td>
<td>89.2</td>
<td>63.6</td>
<td>91.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.2/</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>91.3</td>
<td></td>
</tr>
<tr>
<td><strong>Ensembles on test (from leaderboard as of July 25, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALICE</td>
<td>88.2/</td>
<td>95.7</td>
<td>90.7</td>
<td>83.5</td>
<td>95.2</td>
<td>92.6</td>
<td>68.6</td>
<td>91.1</td>
<td>80.8</td>
<td>86.3</td>
</tr>
<tr>
<td>MT-DNN</td>
<td>87.9/</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
<td>87.6</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.2/</td>
<td>98.6</td>
<td>90.3</td>
<td>86.3</td>
<td>96.8</td>
<td>93.0</td>
<td>67.8</td>
<td>91.6</td>
<td>90.4</td>
<td>88.4</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.8/</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
<td>88.5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>MNLI</th>
<th>QNLI</th>
<th>QQP</th>
<th>RTE</th>
<th>SST-2</th>
<th>MRPC</th>
<th>CoLA</th>
<th>STS-B</th>
<th>WNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT [2]</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td></td>
</tr>
<tr>
<td>RoBERTa [21]</td>
<td>90.2/90.2</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>-</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.8/90.8</td>
<td>94.9</td>
<td>92.3</td>
<td>85.9</td>
<td>97.0</td>
<td>90.8</td>
<td>69.0</td>
<td>92.5</td>
<td>-</td>
</tr>
<tr>
<td><strong>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-DNN* [20]</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
</tr>
<tr>
<td>RoBERTa* [21]</td>
<td>90.8/90.2</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
</tr>
<tr>
<td>XLNet*</td>
<td>90.9/90.9</td>
<td>99.0</td>
<td>90.4</td>
<td>88.5</td>
<td>97.1</td>
<td>92.9</td>
<td>70.2</td>
<td>93.0</td>
<td>92.5</td>
</tr>
<tr>
<td>Model</td>
<td>MNL</td>
<td>QNL</td>
<td>QQP</td>
<td>RTE</td>
<td>SST</td>
<td>MRPC</td>
<td>CoLA</td>
<td>STS</td>
<td>WNLI</td>
</tr>
<tr>
<td>-----------------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
<td>------</td>
<td>------</td>
<td>-----</td>
<td>------</td>
</tr>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td>-</td>
</tr>
<tr>
<td>XLNet&lt;sub&gt;LARGE&lt;/sub&gt;</td>
<td>89.8/-</td>
<td>93.9</td>
<td>91.8</td>
<td>83.8</td>
<td>95.6</td>
<td>89.2</td>
<td>63.6</td>
<td>91.8</td>
<td>-</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.2/90.2</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>91.3</td>
</tr>
<tr>
<td><strong>Ensembles on test (from leaderboard as of July 25, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALICE</td>
<td>88.2/87.9</td>
<td>95.7</td>
<td>90.7</td>
<td>83.5</td>
<td>95.2</td>
<td>92.6</td>
<td>68.6</td>
<td>91.1</td>
<td>80.8</td>
</tr>
<tr>
<td>MT-DNN</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.2/89.8</td>
<td>98.6</td>
<td>90.3</td>
<td>86.3</td>
<td>96.8</td>
<td>93.0</td>
<td>67.8</td>
<td>91.6</td>
<td>90.4</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.8/90.2</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>MNL</th>
<th>QNL</th>
<th>QQP</th>
<th>RTE</th>
<th>SST-2</th>
<th>MRPC</th>
<th>CoLA</th>
<th>STS-B</th>
<th>WNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT [2]</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td>-</td>
</tr>
<tr>
<td>RoBERTa [21]</td>
<td>90.2/90.2</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>-</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.8/90.2</td>
<td>94.9</td>
<td>92.3</td>
<td>85.9</td>
<td>97.0</td>
<td>90.8</td>
<td>69.0</td>
<td>92.5</td>
<td>-</td>
</tr>
<tr>
<td><strong>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-DNN* [20]</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
</tr>
<tr>
<td>RoBERTa* [21]</td>
<td>90.8/90.2</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
</tr>
<tr>
<td>XLNet*</td>
<td>90.9/90.9</td>
<td>99.0</td>
<td>90.4</td>
<td>88.5</td>
<td>97.1</td>
<td>92.9</td>
<td>70.2</td>
<td>93.0</td>
<td>92.5</td>
</tr>
<tr>
<td></td>
<td>MNLI</td>
<td>QNLI</td>
<td>QQP</td>
<td>RTE</td>
<td>SST</td>
<td>MRPC</td>
<td>CoLA</td>
<td>STS</td>
<td>WNLI</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
<td>------</td>
</tr>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT\textsubscript{LARGE}</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td>-</td>
</tr>
<tr>
<td>XLNet\textsubscript{LARGE}</td>
<td>89.8/-</td>
<td>93.9</td>
<td>91.8</td>
<td>83.8</td>
<td>95.6</td>
<td>89.2</td>
<td>63.6</td>
<td>91.8</td>
<td>-</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.2/90.2</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>91.3</td>
</tr>
<tr>
<td><strong>Ensembles on test (from leaderboard as of July 25, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALICE</td>
<td>88.2/87.9</td>
<td>95.7</td>
<td>90.7</td>
<td>83.5</td>
<td>95.2</td>
<td>92.6</td>
<td>68.6</td>
<td>91.1</td>
<td>80.8</td>
</tr>
<tr>
<td>MT-DNN</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.2/89.8</td>
<td>98.6</td>
<td>90.3</td>
<td>86.3</td>
<td>96.8</td>
<td>93.0</td>
<td>67.8</td>
<td>91.6</td>
<td>90.4</td>
</tr>
<tr>
<td>RoBERTa</td>
<td>90.8/90.2</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>MNLI</th>
<th>QNLI</th>
<th>QQP</th>
<th>RTE</th>
<th>SST-2</th>
<th>MRPC</th>
<th>CoLA</th>
<th>STS-B</th>
<th>WNLI</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Single-task single models on dev</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BERT [2]</td>
<td>86.6/-</td>
<td>92.3</td>
<td>91.3</td>
<td>70.4</td>
<td>93.2</td>
<td>88.0</td>
<td>60.6</td>
<td>90.0</td>
<td>-</td>
</tr>
<tr>
<td>RoBERTa [21]</td>
<td>90.2/90.2</td>
<td>94.7</td>
<td>92.2</td>
<td>86.6</td>
<td>96.4</td>
<td>90.9</td>
<td>68.0</td>
<td>92.4</td>
<td>-</td>
</tr>
<tr>
<td>XLNet</td>
<td>90.8/90.8</td>
<td>94.9</td>
<td>92.3</td>
<td>85.9</td>
<td>97.0</td>
<td>90.8</td>
<td>69.0</td>
<td>92.5</td>
<td>-</td>
</tr>
<tr>
<td><strong>Multi-task ensembles on test (from leaderboard as of Oct 28, 2019)</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MT-DNN* [20]</td>
<td>87.9/87.4</td>
<td>96.0</td>
<td>89.9</td>
<td>86.3</td>
<td>96.5</td>
<td>92.7</td>
<td>68.4</td>
<td>91.1</td>
<td>89.0</td>
</tr>
<tr>
<td>RoBERTa* [21]</td>
<td>90.8/90.2</td>
<td>98.9</td>
<td>90.2</td>
<td>88.2</td>
<td>96.7</td>
<td>92.3</td>
<td>67.8</td>
<td>92.2</td>
<td>89.0</td>
</tr>
<tr>
<td>XLNet*</td>
<td>90.9/90.9 \uparrow</td>
<td>99.0 \uparrow</td>
<td>90.4\uparrow</td>
<td>88.5</td>
<td>97.1\uparrow</td>
<td>92.9</td>
<td>70.2</td>
<td>93.0</td>
<td>92.5</td>
</tr>
</tbody>
</table>
XLNet outperforms RoBERTa on all considered tasks

Almost identical training recipes.
# Ablation Study

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1</td>
<td>EM</td>
<td>m/mm</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BERT-Base</td>
<td>64.3</td>
<td>76.30</td>
<td>84.34/84.65</td>
<td>92.78</td>
</tr>
<tr>
<td>2</td>
<td>DAE + Transformer-XL</td>
<td>65.03</td>
<td>79.56</td>
<td>84.88/84.45</td>
<td>92.60</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base (K = 7)</td>
<td>66.05</td>
<td><strong>81.33</strong></td>
<td><strong>85.84/85.43</strong></td>
<td>92.66</td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base (K = 6)</td>
<td>66.66</td>
<td>80.98</td>
<td>85.63/85.12</td>
<td><strong>93.35</strong></td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.15</td>
<td>85.32/85.05</td>
<td>92.78</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>80.61</td>
<td>85.49/85.02</td>
<td>93.12</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.65</td>
<td>85.31/84.99</td>
<td>92.66</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td><strong>66.76</strong></td>
<td>79.83</td>
<td>85.32/85.09</td>
<td>92.89</td>
</tr>
</tbody>
</table>
## Ablation Study

Transformer-XL and permutation LM contribute to the performance

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI m/mm</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1</td>
<td>EM</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>BERT-Base</td>
<td>64.3</td>
<td>76.30</td>
<td>73.66</td>
<td>84.34/84.65</td>
</tr>
<tr>
<td>2</td>
<td>DAE + Transformer-XL</td>
<td>65.03</td>
<td>79.56</td>
<td>76.80</td>
<td>84.88/84.45</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base ($K = 7$)</td>
<td>66.05</td>
<td>81.33</td>
<td>78.46</td>
<td><strong>85.84/85.43</strong></td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base ($K = 6$)</td>
<td>66.66</td>
<td>80.98</td>
<td>78.18</td>
<td>85.63/85.12</td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.15</td>
<td>77.27</td>
<td>85.32/85.05</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>80.61</td>
<td>77.91</td>
<td>85.49/85.02</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.65</td>
<td>77.87</td>
<td>85.31/84.99</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td><strong>66.76</strong></td>
<td>79.83</td>
<td>76.94</td>
<td>85.32/85.09</td>
</tr>
</tbody>
</table>
Ablation Study

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>BERT-Base</td>
<td>64.3</td>
<td>76.30</td>
<td>84.34/84.65</td>
<td>92.78</td>
</tr>
<tr>
<td>2</td>
<td>DAE + Transformer-XL</td>
<td>65.03</td>
<td>79.56</td>
<td>84.88/84.45</td>
<td>92.60</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base (K = 7)</td>
<td>66.05</td>
<td><strong>81.33</strong></td>
<td><strong>85.84/85.43</strong></td>
<td>92.66</td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base (K = 6)</td>
<td>66.66</td>
<td>80.98</td>
<td>85.63/85.12</td>
<td><strong>93.35</strong></td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.15</td>
<td>85.32/85.05</td>
<td>92.78</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>80.61</td>
<td>85.49/85.02</td>
<td>93.12</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.65</td>
<td>85.31/84.92</td>
<td>92.66</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td><strong>66.76</strong></td>
<td>79.81</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Memory caching is important. RACE involves longest contexts of the 4
# Ablation Study

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>F1</td>
<td>EM</td>
<td>m/mm</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>64.3</td>
<td>76.30</td>
<td>84.34/84.65</td>
<td>92.78</td>
</tr>
<tr>
<td>2</td>
<td>DAE + Transformer-XL</td>
<td>65.03</td>
<td>79.56</td>
<td>84.88/84.45</td>
<td>92.60</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base ($K = 7$)</td>
<td>66.05</td>
<td>81.33</td>
<td>85.84/85.43</td>
<td>92.66</td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base ($K = 6$)</td>
<td>66.66</td>
<td>80.98</td>
<td>85.63/85.12</td>
<td><strong>93.35</strong></td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.15</td>
<td>85.32/85.05</td>
<td>92.78</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>80.61</td>
<td>85.49/85.02</td>
<td>93.12</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.65</td>
<td>85.31/84.99</td>
<td>92.66</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td><strong>66.76</strong></td>
<td>79.83</td>
<td>85.32/85.09</td>
<td>92.89</td>
</tr>
</tbody>
</table>

*Bidirectional context is important*
## Ablation Study

Next-sentence prediction objective does not necessarily help

<table>
<thead>
<tr>
<th>#</th>
<th>Model</th>
<th>RACE</th>
<th>SQuAD2.0</th>
<th>MNLI</th>
<th>SST-2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>F1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>BART + Transformer XL</td>
<td>65.45</td>
<td>76.30</td>
<td>84.34/84.65</td>
<td>92.78</td>
</tr>
<tr>
<td>3</td>
<td>XLNet-Base ($K = 7$)</td>
<td>66.05</td>
<td>79.56</td>
<td>84.88/84.45</td>
<td>92.60</td>
</tr>
<tr>
<td>4</td>
<td>XLNet-Base ($K = 6$)</td>
<td>66.66</td>
<td>81.33</td>
<td>85.84/85.43</td>
<td>92.66</td>
</tr>
<tr>
<td>5</td>
<td>- memory</td>
<td>65.55</td>
<td>80.98</td>
<td>85.63/85.12</td>
<td>93.35</td>
</tr>
<tr>
<td>6</td>
<td>- span-based pred</td>
<td>65.95</td>
<td>78.18</td>
<td>85.32/85.05</td>
<td>92.78</td>
</tr>
<tr>
<td>7</td>
<td>- bidirectional data</td>
<td>66.34</td>
<td>80.15</td>
<td>85.49/85.02</td>
<td>93.12</td>
</tr>
<tr>
<td>8</td>
<td>+ next-sent pred</td>
<td>66.76</td>
<td>79.83</td>
<td>85.31/84.99</td>
<td>92.66</td>
</tr>
</tbody>
</table>
XLNet is
The best pretrained model today
Given standard FLOPs.

![Diagram showing accuracy vs FLOPs for XLNet, ALBERT, T5, RoBERTa, and BERT-Large]
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel*, Noam Shazeer*, Adam Roberts*, Katherine Lee*, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu

A systematic study of pre-training objectives, architectures, datasets, transfer approaches and other factors to create the best architecture. Results: T5 models - Base, Small, Large, 3B, 11B
Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer

Colin Raffel*, Noam Shazeer*, Adam Roberts*, Katherine Lee*, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J. Liu

A systematic study of pre-training objectives, architectures, datasets, transfer approaches and other factors to create the best architecture. Results: T5 models - Base, Small, Large, 3B, 11B
Question Answering on SQuAD1.1 dev

From: paperswithcode.com/sota/
Question Answering on SQuAD1.1 dev

<table>
<thead>
<tr>
<th>RANK</th>
<th>METHOD</th>
<th>EM</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>T5-11B</strong></td>
<td>90.06</td>
<td>95.64</td>
</tr>
<tr>
<td>2</td>
<td>XLNet (no data aug)</td>
<td>88.95</td>
<td>94.52</td>
</tr>
<tr>
<td>3</td>
<td>T5-3B</td>
<td>88.53</td>
<td>94.95</td>
</tr>
<tr>
<td>4</td>
<td>T5-Large</td>
<td>86.66</td>
<td>93.79</td>
</tr>
<tr>
<td>5</td>
<td>T5-Base</td>
<td>85.44</td>
<td>92.08</td>
</tr>
<tr>
<td>6</td>
<td>BERT large (+TriviaQA)</td>
<td>84.2</td>
<td>91.1</td>
</tr>
</tbody>
</table>

From: paperswithcode.com/sota/
Text-to-text framework

"translate English to German: That is good."

"cola sentence: The course is jumping well."

"stsrb sentence1: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."

"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi..."

"Das ist gut."

"not acceptable"

"3.8"

"six people hospitalized after a storm in attala county."
Exploration of unsupervised objectives

High-level approaches:
- Language modeling
- BERT-style
- Deshuffling

Corruption strategies:
- Mask
- Replace spans
- Drop

Corruption rate:
- 10%
- 15%
- 25%
- 50%

Corrupted span length:
- 2
- 3
- 5
- 10
Exploration of unsupervised objectives

**High-level approaches**
- Language modeling
- BERT-style
- Deshuffling

**Corruption strategies**
- Mask
- Replace spans
- Drop

**Corruption rate**
- 10%
- 50%

**Corrupted span length**
- 2
- 10

**In:** Thank you <X> me to your party <Y> week
**Out:** <X> for inviting <Y> last <Z>
## Examples of the unsupervised objectives

<table>
<thead>
<tr>
<th>Objective</th>
<th>Inputs</th>
<th>Targets</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix language modeling</td>
<td>Thank you for inviting</td>
<td>me to your party last week.</td>
</tr>
<tr>
<td>BERT-style</td>
<td>Thank you <code>&lt;M&gt;</code> <code>&lt;M&gt;</code> me to your party <code>&lt;apple&gt;</code> week.</td>
<td>(original text)</td>
</tr>
<tr>
<td>Deshuffling</td>
<td>party me for your to . last fun you inviting week Thank you <code>&lt;M&gt;</code> <code>&lt;M&gt;</code> me to your party <code>&lt;M&gt;</code> week.</td>
<td>(original text)</td>
</tr>
<tr>
<td>I.i.d. noise, mask tokens</td>
<td>Thank you <code>&lt;M&gt;</code> <code>&lt;M&gt;</code> me to your party <code>&lt;M&gt;</code> week.</td>
<td>(original text)</td>
</tr>
<tr>
<td>I.i.d. noise, replace spans</td>
<td>Thank you <code>&lt;X&gt;</code> me to your party <code>&lt;Y&gt;</code> week.</td>
<td><code>&lt;X&gt;</code> for inviting <code>&lt;Y&gt;</code> last <code>&lt;Z&gt;</code> for inviting last <code>&lt;X&gt;</code> for inviting me <code>&lt;Y&gt;</code> your party last <code>&lt;Z&gt;</code></td>
</tr>
<tr>
<td>I.i.d. noise, drop tokens</td>
<td>Thank you me to your party week.</td>
<td></td>
</tr>
<tr>
<td>Random spans</td>
<td>Thank you <code>&lt;X&gt;</code> to <code>&lt;Y&gt;</code> week.</td>
<td></td>
</tr>
</tbody>
</table>
Other parameters considered

- Datasets for pre-training
  - Introduce C4: cleaned Common Crawl dataset (745GB)
- Size of the pre-training dataset
- Training strategy
- Scaling: what to do with 4x compute resources
The best model based on experiments: T5

- Encoder-decoder Transformer architecture
- Span-corruption objective, mean span of 3 and corrupt 15%
- Increase number of pre-training steps and batch size
- Use C4 to avoid repetition (large dataset)
- Train 5 different sizes of the models
- Multi-task pre-training
- Fine-tune on individual GLUE/SuperGLUE
- Beam search for tasks with long sequences
Takeaways and insights

Systematic study can lead to an improved model

- Increasing the training time and/or model size improves the baseline
- Objectives that produce short target sequences are more computationally efficient
- Ensembling models that were only fine-tuned separately can give substantial performance and could be a cheaper mean of improving performance
- Pre-training on on-domain data can improve performance
- Updating all parameters during fine-tuning is the most effective but costly
References

Transformer paper: https://arxiv.org/abs/1706.03762
Transformer-XL paper: https://arxiv.org/abs/1
RoBERTa paper: https://arxiv.org/pdf/1907.11692.pdf901.02860
XLNet blog post: https://towardsdatascience.com/what-is-xlnet-and-why-it-outperforms-bert-8d8f2ce710335