Language Modeling with Deep Transformers

Kazuki Irie, Albert Zeyer, Ralf Schlüter, Hermann Ney

Human Language Technology and Pattern Recognition Group
RWTH Aachen University, Aachen, Germany

INTERSPEECH 2019, Graz, Austria
Neural Networks for Language Modeling [Thu-O-10-1], September 19, 2019
Introduction

• 2017: Advent of **Transformer** [Vaswani & Shazeer\(^+\) 17] in NLP/beyond.

• Originally an **encoder-decoder** model for **machine translation**.

• **Decoder** component: **language model**
  – Early work in text generation (5 layers) [Liu & Saleh\(^+\) 18] *ICLR 2018*

• Gain in popularity more recently:
  – Google 64-layer Transformer character LM [Al-Rfou & Choe\(^+\) 19] *AAAI 2019*
  – OpenAI GPT-2 LM (48 layers) [Radford & Wu\(^+\) 19] *Blog February 2019*

• **Large scale language model** pre-training at the center of interest in NLP.
  – Nvidia, Megatron LM (72 layers) *Blog August 2019*
  – Salesforce, Controllable Transformer LM (48 layers) *Last week!*
Contributions of this work

• **Application of Transformer language models to ASR**
  – Successful training of **deep and powerful** Transformer language models.
  – Evaluation in **both hybrid and attention based end-to-end ASR**.
  – Large improvements over the state-of-the-art LSTM LM.

• **Comprehensive hyper-parameter tuning**
  – **Crucial for studying a new model.**
  – In particular for Transformers which have **lots of hyper-parameters**.

• **Demonstration of an LM specific property of Transformers**
  – LM task automatically provides **positional information**: No need for extra signal.

• **Analysis and visualization**

• **Release of model configurations and checkpoints (link in the paper)**

  **Open-source** toolkit RETURNN [Zeyer & Alkhouli† 18]
Transformer Language Model

- Stack $L$ layers; each consisting of self-attention and feed-forward modules.
- Apply residual connections and layer normalization across modules.
- Self-attention typically has multiple attention heads.
Experimental Setups

**LibriSpeech** dataset [Panayotov & Chen¹ 15].
- 960h audio, read speech transcriptions.
- **Large LM task**: 200K vocab, **800M-word** extra textual training data.

Language modeling for **speech recognition** in **2 settings**:

- **Word-level** models for conventional **hybrid** HMM/NN system by **lattice resoring** [Sundermeyer & Tüske¹ 14].
  Push-forwarding **Transformer states** instead LSTM states.

- **BPE subword-level** models for **end-to-end** attention based system by **shallow fusion** [Gülçehre & Firat¹ 17, Toshniwal & Kannan¹ 18].

**Intensive tuning of the baseline LSTM LM** [Sundermeyer & Schlüter¹ 12]
- All tuning details provided in the paper.
- Wide model gave the best results: 2 layers with 4096 LSTM nodes.
- Rel. improvements in PPL over 4-gram of **about 58%**.
Optimization of Transformer models

Exhaustive list of hyper-parameters is long.

- Number of layers & dimension of the residual connection.
- (Dimension of input word embeddings).
- For each layer: number of attention heads, dimension of the key and query, dimension of the value, and dimension of the feed-forward layer.

To reduce this complexity,

- Use the same dimension for key, query, value, and the residual connection.
- Use the same dimensionality across all layers.

4 hyper-parameters to describe all our models.

- Number of layers $L$.
- Feed-forward dimension $d_{ff}$.
- Residual dimension $d_{res}$.
- Number of attention heads $H$. 
Effect of Depth and Width (Highlight)

Perplexity after 2.5 epoch \((H = 8, d_{\text{res}} = 512)\).

<table>
<thead>
<tr>
<th>(L)</th>
<th>(d_{\text{ff}})</th>
<th>Params. in M</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Train</td>
</tr>
<tr>
<td>12</td>
<td>243</td>
<td>67.6</td>
<td>67.1</td>
</tr>
<tr>
<td>24</td>
<td>281</td>
<td>62.2</td>
<td>62.3</td>
</tr>
<tr>
<td>42</td>
<td>338</td>
<td>59.0</td>
<td>59.6</td>
</tr>
<tr>
<td>6</td>
<td>8,192</td>
<td>66.7</td>
<td>66.7</td>
</tr>
<tr>
<td>12</td>
<td>262</td>
<td>63.5</td>
<td>63.8</td>
</tr>
<tr>
<td>4</td>
<td>16,384</td>
<td>67.6</td>
<td>67.4</td>
</tr>
<tr>
<td>32,768</td>
<td>344</td>
<td>65.4</td>
<td>68.4</td>
</tr>
</tbody>
</table>

- For a given parameter budget, **deep models tend to perform better.**

**Full tuning details in the paper!**
- Effect of **number of heads**: *helps up to 16! 8 is already good.*
- Effect of **activation** ReLU, GeLU, GLU: *the standard ReLU is fine!*
- Parameter **tying** (Universal Transformers): *improvements w/o extra params!*
Optimization of Transformer models: Final results

Further scaling up:
Best model: **96-layer model** \((L = 96, d_{ff} = 2048, d_{res} = 512, H = 8)\)
(112-layer model even got slightly better after camera-ready deadline.)

**Final perplexity** on LibriSpeech 200K vocab word level.

<table>
<thead>
<tr>
<th>LM</th>
<th>Param. in M</th>
<th>Perplexity</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Dev</td>
</tr>
<tr>
<td>4-gram</td>
<td>230</td>
<td>146</td>
</tr>
<tr>
<td>LSTM</td>
<td>1048</td>
<td>60</td>
</tr>
<tr>
<td>Transformer</td>
<td>431</td>
<td>54</td>
</tr>
</tbody>
</table>

Large improvements over the **highly optimized LSTM LM**:
- About **11%** relative improvements in PPL.
Do we need extra positional encoding in Transformer LM?

- Amount of information increases at each time step in LM: position signal?
- **Our finding**: External positional encoding unnecessary.
  - Even slight improvements in perplexity w/o positional encoding.
- **Attention in the first layer** (all 8 heads per target word position shown)

With positional encoding

Without positional encoding
Other Layers: 3 categories

Analysis for the 24-layer model which is also valid for deeper models.

- There are **3 functional groups** of layers.

**Bottom layers (2-3): “Blur”**

≈ **Average** over all positions; Bag-of-words. Global info.

Some heads focus on **difficult words**, here **verandah**.

**Mid layers (4-9): “Window”**

Focus on the **local n-gram**.

**Top layers (10-24): “Structured”**

Attend to some **specific patterns**. Feature detector.
Speech Recognition Experiments: Conventional Hybrid System

WERs (%) for **hybrid** systems on **LibriSpeech 960h**.

- The first pass decoding generates **lattices**.
- **Rescore** the lattices (denoted by →) with LSTM or Transformer (Trafo) LM.

<table>
<thead>
<tr>
<th>Language Model</th>
<th>Prm. in M</th>
<th>dev</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>clean</td>
<td>other</td>
<td>clean</td>
<td>other</td>
<td>clean</td>
<td>other</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
<td>WER</td>
<td>PPL</td>
<td>WER</td>
</tr>
<tr>
<td>4-gram → LSTM</td>
<td>230</td>
<td>152</td>
<td>3.4</td>
<td>141</td>
<td>8.3</td>
<td>158</td>
<td>3.8</td>
<td>146</td>
<td>8.8</td>
</tr>
<tr>
<td>→ Transformer</td>
<td>1048</td>
<td>60</td>
<td>2.3</td>
<td>60</td>
<td>5.4</td>
<td>65</td>
<td>2.6</td>
<td>62</td>
<td>5.9</td>
</tr>
<tr>
<td></td>
<td>431</td>
<td>53</td>
<td>2.1</td>
<td>54</td>
<td>5.2</td>
<td>58</td>
<td>2.5</td>
<td>55</td>
<td>5.6</td>
</tr>
<tr>
<td>LSTM → Trafo</td>
<td>-</td>
<td>-</td>
<td>1.9</td>
<td>-</td>
<td>4.5</td>
<td>-</td>
<td>2.3</td>
<td>-</td>
<td>5.0</td>
</tr>
</tbody>
</table>

**Large improvements over the highly optimized LSTM LM:**

- 10% relative improvements in PPL translate to:
- 4% to 10% relative improvements in WER.

**Define new state-of-the-art results** [Lüscher & Beck 19] on LibriSpeech 960h.
Speech Recognition Experiments: Attention-based System

WERs (%) for **attention-based models** on LibriSpeech **960h** dataset. Perplexities are on the **10K BPE** level

<table>
<thead>
<tr>
<th>Language Model</th>
<th>Beam</th>
<th>dev clean PPL</th>
<th>WER</th>
<th>other clean PPL</th>
<th>WER</th>
<th>test clean PPL</th>
<th>WER</th>
<th>test other PPL</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>12</td>
<td>-</td>
<td>4.3</td>
<td>-</td>
<td>12.9</td>
<td>-</td>
<td>4.4</td>
<td>-</td>
<td>13.5</td>
</tr>
<tr>
<td>LSTM</td>
<td>64</td>
<td>44</td>
<td>2.9</td>
<td>46</td>
<td>8.9</td>
<td>47</td>
<td>3.2</td>
<td>47</td>
<td>9.9</td>
</tr>
<tr>
<td>Transformer</td>
<td></td>
<td>36</td>
<td>2.6</td>
<td>39</td>
<td>8.4</td>
<td>39</td>
<td>2.8</td>
<td>39</td>
<td>9.3</td>
</tr>
</tbody>
</table>

- Follow [Hannun & Lee (+) 19] (Interspeech 2019): **Larger beam size** and **end-of-sentence penalty**.
- Again, **large improvements** over the LSTM baseline.
- Best reported WERs for E2E systems **w/o data augmentation** e.g. SpecAugment [Park & Chan (+) 19] (Interspeech 2019).
- Available on: https://github.com/rwth-i6/returnn-experiments
Conclusion

Summary

• Successfully trained deep Transformer LMs with excellent performance for ASR.
• Demonstrated that positional encoding is not needed for Transformer LMs.
• Visualized and identified hierarchical feature engineering inside Transformer language models with link to some fundamental concepts in LM:
  – $N$-gram, bag-of-words, and in top layers; max-entropy model-style features (but data driven)?

Future work

• Further scaling up (layer-wise training).
• Reduce memory requirements of Transformers.
• More study on scalability of Transformer vs. LSTM vs. amount of data.
• For LSTMs: deeper (and wider) models with residual connections and layer normalization e.g., RNMT+ [Chen & Firat+ 18]?
Thank you for your attention.

Thanks to:
Eugen Beck, Liuhui Deng, Christoph Lüscher, Arne Nix, Julian Schamper, and Wei Zhou
References

Character-level language modeling with deeper self-attention.

The best of both worlds: Combining recent advances in neural machine translation.

On using monolingual corpora in neural machine translation.

Sequence-to-sequence speech recognition with time-depth separable convolutions.

Generating wikipedia by summarizing long sequences.
In Int. Conf. on Learning Representations (ICLR), Vancouver, Canada, April 2018.

RWTH ASR systems for LibriSpeech: Hybrid vs Attention.
In Submitted to Interspeech 2019, Graz, Austria, Sept. 2019.

LibriSpeech: an ASR corpus based on public domain audio books.


Language models are unsupervised multitask learners.
References

LSTM neural networks for language modeling.

Lattice decoding and rescoring with long-span neural network language models.


Attention is all you need.

RETURNN as a generic flexible neural toolkit with application to translation and speech recognition.
In Proc. of the Joint Conf. of the 47th Annual Meeting of the ACL and the 4th Int. Joint Conf. on Natural Language Processing of the AFNLP, Melbourne, Australia, July 2018.