Sequence-to-sequence models with attention

A woman is throwing a frisbee in a park.
A dog is standing on a hardwood floor.
A stop sign is on a road with a mountain in the background.
A little girl sitting on a bed with a teddy bear.
A group of people sitting on a boat in the water.
A giraffe standing in a forest with trees in the background.

$f = (La, croissance, économique, s'est, ralentie, ces, dernières, années, .)$

$h_j, a_j, z_j, u_j$
Overview

• Image captioning with attention
• Neural machine translation with attention
  • Recurrent models with global and local attention
  • Google Neural Machine Translation
  • Convolutional sequence to sequence models
  • Attention without recurrence or convolutions
Review: Image captioning

```
CNN

"START"

Classifier

h0

"The"

Classifier

h1

"dog"

Classifier

h2

"is"

Classifier

h3

"hiding"

Classifier

h4

"STOP"

Classifier

h5

"The" "is" "hiding"

Classifier

"dog" "is" "hiding" "STOP"
```
Review: Image captioning

```
The dog is hiding
A yellow cat is sitting on the
One dog is sitting on the
```

CNN

```
“START”
“The”
“dog”
“is”
“hiding”
```
Beam search

- Maintain $k$ top-scoring candidate sentences (according to sum of per-word log-likelihoods)
  - At each step, generate all their successors and reduce to $k$ (*beam width*)
How to evaluate image captioning?

Reference sentences (written by human annotators):

- “A dog hides underneath a bed with its face peeking out of the bed skirt”
- “The small white dog is peeking out from under the bed”
- “A dog is peeking its head out from underneath a bed skirt”
- “A dog peeking out from under a bed”
- “A dog that is under a bed on the floor”

Generated sentence:
- “A dog is hiding”
BLEU: Bilingual Evaluation Understudy

• **N-gram precision**: count the number of n-gram matches between candidate and reference translation, divide by total number of n-grams in reference translation
  • Clip counts by the maximum number of times an n-gram occurs in any reference translation
  • Multiply by *brevity penalty* to penalize short translations

• Most commonly used measure despite well-known shortcomings

## 2015 Captioning Challenge

<table>
<thead>
<tr>
<th>Method</th>
<th>CIDEr-D</th>
<th>Meteor</th>
<th>ROUGE-L</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
</tr>
</thead>
<tbody>
<tr>
<td>m-RNN (Baidu/ UCLA)</td>
<td>0.886</td>
<td>0.238</td>
<td>0.524</td>
<td>0.72</td>
<td>0.553</td>
<td>0.41</td>
<td>0.302</td>
</tr>
<tr>
<td>m-RNN[^15]</td>
<td>0.845</td>
<td>0.250</td>
<td>0.513</td>
<td>0.689</td>
<td>0.515</td>
<td>0.372</td>
<td>0.268</td>
</tr>
<tr>
<td>MSR Captive</td>
<td>0.725</td>
<td>0.204</td>
<td>0.499</td>
<td>0.654</td>
<td>0.498</td>
<td>0.362</td>
<td>0.241</td>
</tr>
<tr>
<td>Google[^4]</td>
<td>0.833</td>
<td>0.231</td>
<td>0.505</td>
<td>0.683</td>
<td>0.51</td>
<td>0.377</td>
<td>0.281</td>
</tr>
<tr>
<td>Berkeley LRCN</td>
<td>0.787</td>
<td>0.233</td>
<td>0.500</td>
<td>0.654</td>
<td>0.498</td>
<td>0.362</td>
<td>0.241</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>0.853</td>
<td>0.252</td>
<td>0.484</td>
<td>0.663</td>
<td>0.469</td>
<td>0.321</td>
<td>0.217</td>
</tr>
<tr>
<td>MSR[^8]</td>
<td>0.865</td>
<td>0.252</td>
<td>0.484</td>
<td>0.663</td>
<td>0.469</td>
<td>0.321</td>
<td>0.217</td>
</tr>
<tr>
<td>Montreal/Toronto[^10]</td>
<td>0.85</td>
<td>0.243</td>
<td>0.513</td>
<td>0.689</td>
<td>0.515</td>
<td>0.372</td>
<td>0.268</td>
</tr>
<tr>
<td>PicSOM[^13]</td>
<td>0.833</td>
<td>0.231</td>
<td>0.505</td>
<td>0.683</td>
<td>0.51</td>
<td>0.377</td>
<td>0.281</td>
</tr>
<tr>
<td>Tsinghua Bigeye[^14]</td>
<td>0.673</td>
<td>0.207</td>
<td>0.49</td>
<td>0.671</td>
<td>0.494</td>
<td>0.35</td>
<td>0.241</td>
</tr>
<tr>
<td>MBL[^7]</td>
<td>0.74</td>
<td>0.219</td>
<td>0.499</td>
<td>0.666</td>
<td>0.498</td>
<td>0.362</td>
<td>0.241</td>
</tr>
<tr>
<td>Human[^5]</td>
<td>0.854</td>
<td>0.252</td>
<td>0.484</td>
<td>0.663</td>
<td>0.469</td>
<td>0.321</td>
<td>0.217</td>
</tr>
</tbody>
</table>

### Metrics

- **CIDEr-D**: Consensus-based Image Description Evaluation
- **METEOR**: Meteor Universal: Language Specific Translation Evaluation for Any Target Language
- **ROUGE-L**: ROUGE: A Package for Automatic Evaluation of Summaries
- **BLEU**: BLEU: a Method for Automatic Evaluation of Machine Translation
### 2015 Captioning Challenge

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Human</td>
<td>0.638</td>
<td>0.675</td>
<td>4.836</td>
<td>3.428</td>
<td>0.352</td>
</tr>
<tr>
<td>Google</td>
<td>0.676</td>
<td>0.667</td>
<td>4.417</td>
<td>3.746</td>
<td>0.356</td>
</tr>
<tr>
<td>MSR</td>
<td>0.578</td>
<td>0.675</td>
<td>4.417</td>
<td>2.716</td>
<td>0.320</td>
</tr>
<tr>
<td>Montreal</td>
<td>0.540</td>
<td>0.561</td>
<td>4.246</td>
<td>3.453</td>
<td>0.290</td>
</tr>
<tr>
<td>MSR Camera</td>
<td>0.638</td>
<td>0.675</td>
<td>4.836</td>
<td>3.428</td>
<td>0.352</td>
</tr>
<tr>
<td>Berkeley</td>
<td>0.636</td>
<td>0.674</td>
<td>4.826</td>
<td>3.428</td>
<td>0.352</td>
</tr>
<tr>
<td>m-RNN</td>
<td>0.634</td>
<td>0.673</td>
<td>4.816</td>
<td>3.428</td>
<td>0.352</td>
</tr>
<tr>
<td>Nearest Neighbor</td>
<td>0.216</td>
<td>0.255</td>
<td>3.801</td>
<td>2.716</td>
<td>0.196</td>
</tr>
<tr>
<td>PicSOM</td>
<td>0.202</td>
<td>0.250</td>
<td>3.965</td>
<td>2.552</td>
<td>0.182</td>
</tr>
<tr>
<td>Brno University</td>
<td>0.194</td>
<td>0.213</td>
<td>3.079</td>
<td>3.482</td>
<td>0.154</td>
</tr>
<tr>
<td>m-RNN (Baidu/ UCLA)</td>
<td>0.190</td>
<td>0.241</td>
<td>3.831</td>
<td>2.548</td>
<td>0.195</td>
</tr>
<tr>
<td>MIL</td>
<td>0.168</td>
<td>0.197</td>
<td>3.349</td>
<td>2.915</td>
<td>0.159</td>
</tr>
<tr>
<td>MLBL</td>
<td>0.167</td>
<td>0.196</td>
<td>3.659</td>
<td>2.420</td>
<td>0.156</td>
</tr>
</tbody>
</table>

**Legend**

- **M1**: Percentage of captions that are evaluated as better or equal to human caption.
- **M2**: Percentage of captions that pass the Turing Test.
- **M3**: Average correctness of the captions on a scale 1-5 (incorrect - correct).
- **M4**: Average amount of detail of the captions on a scale 1-5 (lack of details - very detailed).
- **M5**: Percentage of captions that are similar to human description.
Captioning with attention

K. Xu et al., *Show, Attend and Tell: Neural Image Caption Generation with Visual Attention*, ICML 2015
Captioning with attention

Image

CNN

Feature map 14x14x512

Attention map over locations

α₁

α₂

p(y₁)

h₀

h₁

z₁

y₀

<START>

zₜ = ϕ(α, αₜ)

Context vector determined by feature map and attention

Adapted from Stanford CS231n, Berkeley CS294
Captioning with attention

Adapted from Stanford CS231n, Berkeley CS294
Captioning with attention

Adapted from Stanford CS231n, Berkeley CS294

Image

CNN

Feature map 14x14x512

$h_0$  $h_1$  $h_2$

$\alpha_1$  $\alpha_2$  $\alpha_3$

$p(y_1)$  $p(y_2)$

$z_1$  $y_0$  $z_2$  $y_1$

<START>
“Soft” and “hard” attention

**Soft attention:**
Average over locations of feature map weighted by attention: \( z_t = \sum_i \alpha_{i,t} a_i \)

Train with gradient descent

**Hard attention:**
Sample ONE location: \( z_t \) is a random variable taking on values \( a_i \) with probabilities \( \alpha_{i,t} \)

Can’t use gradient descent; need reinforcement learning

Adapted from Berkeley CS294
“Soft” and “hard” attention

**Soft attention:**
Average over locations of feature map weighted by attention: \( z_t = \sum_i \alpha_{i,t} a_i \)

**Hard attention:**
Sample ONE location: \( z_t \) is a random variable taking on values \( a_i \) with probabilities \( \alpha_{i,t} \)
## Results

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Model</th>
<th>BLEU-1</th>
<th>BLEU-2</th>
<th>BLEU-3</th>
<th>BLEU-4</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flickr8k</td>
<td>Google NIC</td>
<td>63</td>
<td>41</td>
<td>27</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>67</td>
<td>44.8</td>
<td>29.9</td>
<td>19.5</td>
<td>18.93</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>67</td>
<td>45.7</td>
<td>31.4</td>
<td>21.3</td>
<td>20.30</td>
</tr>
<tr>
<td>Flickr30k</td>
<td>Google NIC</td>
<td>66.3</td>
<td>42.3</td>
<td>27.7</td>
<td>18.3</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>66.7</td>
<td>43.4</td>
<td>28.8</td>
<td>19.1</td>
<td>18.49</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>66.9</td>
<td>43.9</td>
<td>29.6</td>
<td>19.9</td>
<td>18.46</td>
</tr>
<tr>
<td>COCO</td>
<td>Google NIC</td>
<td>66.6</td>
<td>46.1</td>
<td>32.9</td>
<td>24.6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Soft-Attention</td>
<td>70.7</td>
<td>49.2</td>
<td>34.4</td>
<td>24.3</td>
<td>23.90</td>
</tr>
<tr>
<td></td>
<td>Hard-Attention</td>
<td>71.8</td>
<td>50.4</td>
<td>35.7</td>
<td>25.0</td>
<td>23.04</td>
</tr>
</tbody>
</table>
Example Results

• Good captions

A woman is throwing a **frisbee** in a park.

A **dog** is standing on a hardwood floor.

A **stop** sign is on a road with a mountain in the background.

A little girl sitting on a bed with a teddy bear.

A group of **people** sitting on a boat in the water.

A giraffe standing in a forest with trees in the background.
Example Results

- Mistakes

A large white **bird** standing in a forest.

A woman holding a **clock** in her hand.

A man wearing a hat and a hat on a **skateboard**.

A person is standing on a beach with a **surfboard**.

A woman is sitting at a table with a large **pizza**.

A man is talking on his cell **phone** while another man watches.
Machine translation: Vanilla Seq2Seq

I. Sutskever, O. Vinyals, Q. Le, *Sequence to Sequence Learning with Neural Networks*, NIPS 2014

Machine translation with attention

- Key idea: translation requires *alignment*

What is the anticipated cost of collecting fees under the new proposal?

En vertu des nouvelles propositions, quel est le coût prévu de perception des droits?
Machine translation with attention

Standard encoder-decoder

Context vector \( c = h_T \) is used for decoding each word.

Attention-based model

Context vector \( c_t \) pays attention to different phrases in the source when generating each word.

Global attentional model

Alignment model:
\[ e_{tj} = a(s_{t-1}, h_j) \]
\[ \alpha_{tj} = \text{softmax}(e_{tj}) \]

Bidirectional LSTM encoder:
\[ h_j = [\vec{h}_j; \tilde{h}_j] \]

Context vector:
\[ c_t = \sum_{j=1}^{T} \alpha_{tj} h_j \]
Example alignment
Quantitative evaluation

<table>
<thead>
<tr>
<th></th>
<th>No attention</th>
<th>With attention (trained with sentence length &lt;= 30)</th>
<th>With attention (trained with sentence length &lt;= 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>RNNsearch-50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNNsearch-30</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNNenc-50</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNNenc-30</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Local attention

- Key idea: design mechanism similar to “hard attention” but differentiable
- For each target word, predict an aligned position $p_t$ in the source; form context from fixed-size window around $p_t$

M.-T. Luong, H. Pham, and C. Manning, *Effective approaches to attention-based neural machine translation*, EMNLP 2015
Local attention

Predictive distribution
Attentional hidden state

Attention Layer

Context vector
Aligned position
Local weights

\[ p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t)) \]

length of input sequence
Local attention

\[ p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t)) \]

\[ \alpha_{tj} = \text{softmax}(h_j^T W_a s_t) \exp\left(-\frac{(j - p_t)^2}{2\sigma^2}\right) \]
Local attention

\[ c_t = \sum_j \alpha_{tj} h_j \]

\[ \alpha_{tj} = \text{softmax}(h_j^T W_a s_t) \exp\left(-\frac{(j - p_t)^2}{2\sigma^2}\right) \]

\[ p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t)) \]
Local attention

\[ c_t = \sum_j \alpha_{tj} h_j \]

\[ \alpha_{tj} = \text{softmax}(h_j^T W_a s_t) \exp\left(-\frac{(j - p_t)^2}{2\sigma^2}\right) \]

\[ \tilde{s}_t = \tanh(W_c [c_t; s_t]) \]

\[ p_t = L_s \cdot \sigma(v_p^T \tanh(W_p s_t)) \]
Local attention

\[ p(y_t | y_{<t}, x) = \text{softmax}(W_s \tilde{s}_t) \]

\[ \tilde{s}_t = \tanh(W_c [c_t; s_t]) \]

\[ c_t = \sum_j \alpha_{tj} h_j \]

\[ \alpha_{tj} = \text{softmax}(h_j^T W_a s_t) \exp(-\frac{(j - p_t)^2}{2\sigma^2}) \]
Results

- English-German translation
Google Neural Machine Translation (GNMT)

Google’s Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi
yonghui,schuster,zhifengc,qvl,mnorouzi@google.com
Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Łukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, Jeffrey Dean

Y. Wu et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv 2016

https://www.nytimes.com/2016/12/14/magazine/the-great-ai-awakening.html
Google Neural Machine Translation (GNMT)

Y. Wu et al., Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, arXiv 2016
Google Neural Machine Translation (GNMT)

- **Encoder**
  - Model is partitioned 8 ways and is placed on 8 different GPUs
  - The bottom bi-directional encoder layers compute in parallel first
  - Once both finish, the uni-directional encoder layers can start computing, each on a separate GPU
  - Uses stacked LSTMs with residual connections
Google Neural Machine Translation (GNMT)

Regular stacked LSTM

Stacked LSTM with residual connections
Google Neural Machine Translation (GNMT)

- **Decoder**
  - Only the bottom decoder layer output is used for obtaining attention context, which is sent directly to all the higher decoder layers
  - The softmax layer is also partitioned and placed on multiple GPUs
Google Neural Machine Translation (GNMT)

- **Standard training objective:** maximize log-likelihood of ground truth output given input:
  \[ \sum_{i} \log P_{W}(Y_{i}^{*} | X_{i}) \]

  - Not related to task-specific reward function (e.g., BLEU score)
  - Does not encourage “better” predicted sentences to get better likelihood

- **GNMT objective:** expectation of rewards over possible predicted sentences \( Y \):
  \[ \sum_{i} \sum_{Y} P_{W}(Y | X_{i}) r(Y, Y_{i}^{*}) \]

  - Use variant of BLEU score to compute reward
  - Reward is not differentiable -- need reinforcement learning to train (initialize with ML-trained model)
Google Neural Machine Translation (GNMT)

- Results on production data (500 randomly sampled sentences from Wikipedia and news websites)

### Table 10: Mean of side-by-side scores on production data

<table>
<thead>
<tr>
<th></th>
<th>PBMT</th>
<th>GNMT</th>
<th>Human</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>English → Spanish</td>
<td>4.885</td>
<td>5.428</td>
<td>5.550</td>
<td>87%</td>
</tr>
<tr>
<td>English → French</td>
<td>4.932</td>
<td>5.295</td>
<td>5.496</td>
<td>64%</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>4.035</td>
<td>4.594</td>
<td>4.987</td>
<td>58%</td>
</tr>
<tr>
<td>Spanish → English</td>
<td>4.872</td>
<td>5.187</td>
<td>5.372</td>
<td>63%</td>
</tr>
<tr>
<td>French → English</td>
<td>5.046</td>
<td>5.343</td>
<td>5.404</td>
<td>83%</td>
</tr>
<tr>
<td>Chinese → English</td>
<td>3.694</td>
<td>4.263</td>
<td>4.636</td>
<td>60%</td>
</tr>
</tbody>
</table>

**Side-by-side scores:** range from 0 (“completely nonsense translation”) to 6 (“perfect translation”), produced by human raters fluent in both languages

**PBMT:** Translation by phrase-based statistical translation system used by Google

**GNMT:** Translation by our GNMT system

**Human:** Translation by humans fluent in both languages
Convolutional sequence models

- Instead of recurrent networks, use 1D convolutional networks
Convolutional sequence models

• From the conclusion:

The preeminence enjoyed by recurrent networks in sequence modeling may be largely a vestige of history. Until recently, before the introduction of architectural elements such as dilated convolutions and residual connections, convolutional architectures were indeed weaker. Our results indicate that with these elements, a simple convolutional architecture is more effective across diverse sequence modeling tasks than recurrent architectures such as LSTMs. Due to the comparable clarity and simplicity of TCNs, we conclude that convolutional networks should be regarded as a natural starting point and a powerful toolkit for sequence modeling.

Convolutional sequence to sequence learning

J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. Dauphin, Convolutional sequence to sequence learning, ICML 2017
Convolutional sequence to sequence learning

- Results

<table>
<thead>
<tr>
<th>WMT’14 English-German</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luong et al. (2015) LSTM (Word 50K)</td>
<td>20.9</td>
</tr>
<tr>
<td>Kalchbrenner et al. (2016) ByteNet (Char)</td>
<td>23.75</td>
</tr>
<tr>
<td>Wu et al. (2016) GNMT (Word 80K)</td>
<td>23.12</td>
</tr>
<tr>
<td>Wu et al. (2016) GNMT (Word pieces)</td>
<td>24.61</td>
</tr>
<tr>
<td>ConvS2S (BPE 40K)</td>
<td>25.16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>WMT’14 English-French</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wu et al. (2016) GNMT (Word 80K)</td>
<td>37.90</td>
</tr>
<tr>
<td>Wu et al. (2016) GNMT (Word pieces)</td>
<td>38.95</td>
</tr>
<tr>
<td>Wu et al. (2016) GNMT (Word pieces) + RL</td>
<td>39.92</td>
</tr>
<tr>
<td>ConvS2S (BPE 40K)</td>
<td>40.51</td>
</tr>
</tbody>
</table>
Attention is all you need

• NMT architecture using only FC layers and attention
  • More efficient and parallelizable than recurrent or convolutional architectures, faster to train, better accuracy

A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, NIPS 2017
Attention is all you need

- NMT architecture using only FC layers and attention

**Encoder:** receives entire input sequence and outputs encoded sequence of the same length

**Decoder:** predicts one word at a time, conditioned on encoder output and previously predicted words

Self-attention
Recall: Self-attention for images

- Used in BigGAN to capture spatial structure

Transformer architecture in detail
Positional encoding

- Hand-crafted encoding (using sines and cosines) is added to every input vector
The decoder generates a **query** describing what it wants to focus on.

**Compute dot products** between the query and the **keys** associated with encoder input, describing how well the keys match the query.

**Sum the values** associated with encoder inputs weighted by the attention weights.

Feed the scores into a softmax to create the attention weights.

Encoder

Decoder
Attention mechanism

- **Scaled dot product attention:**
  \[
  \text{Attention}(Q, K, V) = \text{softmax} \left( \frac{QK^T}{\sqrt{d_k}} \right) V
  \]

- \(Q, K, V\) are matrices with rows corresponding to queries, keys, and values, \(d_k\) is the dim. of the keys.
Attention mechanism

• *Scaled dot product attention:*

\[
\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V
\]

• \(Q, K, V\) are matrices with rows corresponding to queries, keys, and values, \(d_k\) is the dim. of the keys

• *Multi-head attention:* run \(h\) attention models in parallel on top of different linearly projected versions of \(Q, K, V\); concatenate and linearly project the results
Attention mechanism

- **Encoder-decoder attention**: queries come from previous decoder layer, keys and values come from output of encoder
- **Encoder self-attention**: queries, keys, and values come from previous layer of encoder
- **Decoder self-attention**: values corresponding to future outputs are masked out
Results

English German Translation quality

- GNMT (RNN)
- ConvS2S (CNN)
- SlicesNet (CNN)
- Transformer

English French Translation Quality

- GNMT (RNN)
- ConvS2S (CNN)
- Transformer

Parting thoughts

• Methodology for text generation problems
  • Evaluation is tricky
  • Maximum likelihood training is not the most appropriate (but alternatives involve optimizing non-differentiable objectives)

• Attention appears to be a game-changer for NMT (for image captioning, not as much)
  • But there is much more to MT than attention (dealing with unknown words, etc.)

• Recurrent architectures are not the only option for sequence modeling
  • Convolutional and feedforward alternatives should be considered