Copy/Pointer + Transformer

Wei Xu

(many slides from Greg Durrett)
This Lecture

- Sequence-to-Sequence Model
- Attention Mechanism
- Copy Mechanism
- Transformer Architecture
Recap: Seq2Seq Model

- **Encoder**: consumes sequence of tokens, produces a vector. Analogous to encoders for classification/tagging tasks. 

  \[ P(y_i|x, y_1, \ldots, y_{i-1}) = \text{softmax}(W\tilde{h}_i) \]

- **Decoder**: separate module, single cell. Takes two inputs: hidden state (vector \( h \) or tuple \((h, c)\)) and previous token. Outputs token + new state.
At each decoder state, compute a distribution over source inputs based on current decoder state.

Use that in output layer.
Attention

- For each decoder state, compute weighted sum of input states:

\[ c_i = \sum_j \alpha_{ij} h_j \]

\[ e_{ij} = f(\bar{h}_i, h_j) \]

\[ \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})} \]

- No attn:

\[ P(y_i|x,y_1,\ldots,y_{i-1}) = \text{softmax}(W\bar{h}_i) \]

\[ P(y_i|x,y_1,\ldots,y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i]) \]

- Weighted sum of input hidden states (vector):

\[ P(y_i|x,y_1,\ldots,y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i]) \]

- Some function f (next slide):
Attention

Note that this all uses outputs of hidden layers

\[ c_i = \sum_j \alpha_{ij} h_j \]
\[ \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{j'} \exp(e_{ij'})} \]
\[ e_{ij} = f(\bar{h}_i, h_j) \]

- \( f(\bar{h}_i, h_j) = \tanh(W[\bar{h}_i, h_j]) \)
  
  - Bahdanau+ (2014): additive

- \( f(\bar{h}_i, h_j) = \bar{h}_i \cdot h_j \)
  
  - Luong+ (2015): dot product

- \( f(\bar{h}_i, h_j) = \bar{h}_i^\top W h_j \)
  
  - Luong+ (2015): bilinear
What can attention do?

- Learning to subsample tokens

![Diagram](image)

- Need to count (for ordering) and also determine which tokens are in/out

- Content-based addressing

Luong et al. (2015)
Copying Input/Pointers
Want to be able to copy named entities like Pont-de-Buis

\[
P(y_i|x, y_1, \ldots, y_{i-1}) = \text{softmax}(W[c_i; \tilde{h}_i])
\]

from attention

from RNN hidden state

Problem: target word has to be in the vocabulary, attention + RNN need to generate good embedding to pick it

Jean et al. (2015), Luong et al. (2015)
en: The **écotax** portico in *Pont-de-Buis*, … [truncated] …

fr: Le **portique écotaxe** de *Pont-de-Buis*, … [truncated]

nn: Le *unk* de *unk à unk*, … [truncated] …, a été pris

- Vocabulary contains “normal” vocab as well as words in input. Normalizes over both of these:

\[
P(y_i = w | x, y_1, \ldots, y_{i-1}) \propto \begin{cases} 
\exp W_w [c_i; \bar{h}_i] & \text{if } w \text{ in vocab} \\
\exp h_j^T V \bar{h}_i & \text{if } w = x_j 
\end{cases}
\]

- Bilinear function of input representation + output hidden state
Pointer Network

- Standard decoder ($P_{\text{vocab}}$): softmax over vocabulary

$$P(y_i|x, y_1, \ldots, y_{i-1}) = \text{softmax}(W[c_i; \bar{h}_i])$$

- Pointer network ($P_{\text{pointer}}$): predict from source words, instead of target vocabulary

$$P_{\text{pointer}}(y_i|x, y_1, \ldots, y_{i-1}) \propto \begin{cases} h_j^T V \bar{h}_i & \text{if } y_i = w_j \\ 0 & \text{otherwise} \end{cases}$$
Define the decoder model as a mixture model of $P_{\text{vocab}}$ and $P_{\text{pointer}}$

$$P(y_i | x, y_1, \ldots, y_{i-1}) = P(\text{copy})P_{\text{pointer}} + (1 - P(\text{copy}))P_{\text{vocab}}$$

- Predict $P(\text{copy})$ based on decoder state, input, etc.
- Marginalize over copy variable during training and inference
- Model will be able to both generate and copy, flexibly adapt between the two

Gulcehre et al. (2016), Gu et al. (2016)
Copying in Summarization

See et al. (2017)
### Copying in Summarization

<table>
<thead>
<tr>
<th></th>
<th>ROUGE</th>
<th>METEOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>abstractive model</td>
<td>35.46</td>
<td>13.30</td>
</tr>
<tr>
<td>(Nallapati et al., 2016)*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq-to-seq + attn baseline</td>
<td>30.49</td>
<td>11.17</td>
</tr>
<tr>
<td>(150k vocab)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>seq-to-seq + attn baseline</td>
<td>31.33</td>
<td>11.81</td>
</tr>
<tr>
<td>(50k vocab)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>pointer-generator</td>
<td>36.44</td>
<td>15.66</td>
</tr>
<tr>
<td>pointer-generator + coverage</td>
<td><strong>39.53</strong></td>
<td><strong>17.28</strong></td>
</tr>
<tr>
<td>lead-3 baseline (ours)</td>
<td>40.34</td>
<td>17.70</td>
</tr>
<tr>
<td>lead-3 baseline</td>
<td>39.2</td>
<td>15.7</td>
</tr>
<tr>
<td>(Nallapati et al., 2017)*</td>
<td></td>
<td></td>
</tr>
<tr>
<td>extractive model</td>
<td>39.6</td>
<td>16.2</td>
</tr>
<tr>
<td>(Nallapati et al., 2017)*</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See et al. (2017)
Original Text (truncated): lagos, nigeria (cnn) a day after winning nigeria’s presidency, **muhmmadu buhari** told cnn’s christiane amanpour that he plans to aggressively fight corruption that has long plagued nigeria and go after the root of the nation’s unrest. **buhari** said he’ll “rapidly give attention” to curbing violence in the northeast part of nigeria, where the terrorist group boko haram operates. by cooperating with neighboring nations chad, cameroon and niger, he said his administration is confident it will be able to thwart criminals and others contributing to nigeria’s instability. for the first time in nigeria’s history, the opposition defeated the ruling party in democratic elections. **buhari** defeated incumbent goodluck jonathan by about 2 million votes, according to nigeria’s independent national electoral commission. the win comes after a long history of military rule, coups and botched attempts at democracy in africa’s most populous nation.

Figure 1: Comparison of output of 3 abstractive summarization models on a news article. The baseline model makes **factual errors**, a **nonsensical sentence** and struggles with OOV words **muhmmadu buhari**. The pointer-generator model is accurate but **repeats itself**. Coverage eliminates repetition. The final summary is composed from **several fragments**.

Baseline Seq2Seq + Attention: **UNK UNK** says his administration is confident it will be able to **destabilize nigeria’s economy**. **UNK** says his administration is confident it will be able to thwart criminals and other **nigerians**. he says the country has long nigeria and nigeria’s economy.

Pointer-Gen: **muhmmadu buhari** says he plans to aggressively fight corruption in the northeast part of nigeria. he says he’ll “rapidly give attention” to curbing violence in the northeast part of nigeria. he says his administration is confident it will be able to thwart criminals.

Pointer-Gen + Coverage: **muhmmadu buhari** says he plans to aggressively fight corruption that has long plagued nigeria. he says his administration is confident it will be able to thwart criminals. the win comes after a long history of military rule, coups and botched attempts at democracy in africa’s most populous nation.

See et al. (2017)
Transformers
Attention is All You Need

Ashish Vaswani*Google Brain
avaswani@google.com

Noam Shazeer*Google Brain
noam@google.com

Niki Parmar*Google Research
nikip@google.com

Jakob Uszkoreit*Google Research
usz@google.com

Llion Jones*Google Research
llion@google.com

Aidan N. Gomez†University of Toronto
aidan@gcs.toronto.edu

Lukasz Kaiser†Google Brain
lukasz.kaiser@google.com

Illia Polosukhin†

Abstract

The dominant sequence transduction models are based on complex recurrent or convolutional neural networks that include an encoder and a decoder. The best performing models also connect the encoder and decoder through an attention mechanism. We propose a new simple network architecture, the Transformer, based solely on attention mechanisms, dispensing with recurrence and convolutions entirely. Experiments on two machine translation tasks show these models to be superior in quality while being more parallelizable and requiring significantly less time to train. One model achieves 28.4 BLEU on the WMT 2014 English-to-German translation task, improving over the existing best results, including ensembles, by over 2 BLEU. On the WMT 2014 English-to-French translation task, our model establishes a new single-model state-of-the-art BLEU score of 41.8 after training for 3.5 days on eight GPUs, a small fraction of the training costs of the best models from the literature. We show that the Transformer generalizes well to other tasks by applying it successfully to English constituency parsing both with large and limited training data.
Sentence Encoders

- LSTM abstraction: maps each vector in a sentence to a new, context-aware vector

- CNNs do something similar with filters

- Attention can give us a third way to do this

Vaswani et al. (2017)
Self-Attention

- Assume we’re using GloVe — what do we want our neural network to do?

  The ballerina is very excited that she will dance in the show.

- What words need to be contextualized here?
  - Pronouns need to look at antecedents
  - Ambiguous words should look at context
  - Words should look at syntactic parents/children

- Problem: LSTMs and CNNs don’t do this

Vaswani et al. (2017)
The ballerina is very excited that she will dance in the show.

LSTMs/CNNs: tend to look at local context

To appropriately contextualize embeddings, we need to pass information over long distances dynamically for each word

Vaswani et al. (2017)
Self-Attention

- Each word forms a “query” which then computes attention over each word

\[ \alpha_{i,j} = \text{softmax}(x_i^\top x_j) \quad \text{scalar} \]

\[ x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector = sum of scalar } \times \text{ vector} \]

- Multiple “heads” analogous to different convolutional filters. Use parameters \( W_k \) and \( V_k \) to get different attention values + transform vectors

\[ \alpha_{k,i,j} = \text{softmax}(x_i^\top W_k x_j) \quad x'_{k,i} = \sum_{j=1}^{n} \alpha_{k,i,j} V_k x_j \]

Vaswani et al. (2017)
What can self-attention do?

*The ballerina is very excited that *she* will dance in the *show.*

- Attend nearby + to semantically related terms

- Why multiple heads? Softmaxes end up being peaked, single distribution cannot easily put weight on multiple things

Vaswani et al. (2017)
Additional Readings

- "The Illustrated Transformer" by Jay Lamar
  http://jalammar.github.io/illustrated-transformer/

- "The Annotated Transformer" by Sasha Rush
  https://nlp.seas.harvard.edu/2018/04/03/attention.html
Transformer Uses

- **Supervised:** transformer can replace LSTM as encoder, decoder, or both; such as in machine translation and natural language generation tasks.

- **Encoder and decoder are both transformers**

- **Decoder** consumes the previous generated token (and attends to input), but has *no recurrent state*

- **Many other details to get it to work:** residual connections, layer normalization, positional encoding, optimizer with learning rate schedule, label smoothing ....

Vaswani et al. (2017)
Transformers

- Augment word embedding with position embeddings, each dim is a sine/cosine wave of a different frequency. Closer points = higher dot products.
- Works essentially as well as just encoding position as a one-hot vector.

Vaswani et al. (2017)
Residual Connections

- allow gradients to flow through a network directly, without passing through non-linear activation functions
Transformers

- Adam optimizer with varied learning rate over the course of training
- Linearly increase for warmup, then decay proportionally to the inverse square root of the step number
- This part is very important!

Vaswani et al. (2017)
Label Smoothing

- Instead of using a one-hot target distribution, create a distribution that has “confidence” of the correct word and the rest of the “smoothing” mass distributed throughout the vocabulary.

- Implemented by minimizing KL-divergence between smoothed ground truth probabilities and the probabilities computed by model.

\[ I \text{ went to class and took } \underline{_____} \]

<table>
<thead>
<tr>
<th></th>
<th>cats</th>
<th>TV</th>
<th>notes</th>
<th>took</th>
<th>sofa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>0.025</td>
<td>0.025</td>
<td>0.9</td>
<td>0.025</td>
<td>0.025</td>
</tr>
</tbody>
</table>

\[\text{-------- with label smoothing}\]
Transformers

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EN-DE</td>
<td>EN-FR</td>
</tr>
<tr>
<td>ByteNet [18]</td>
<td>23.75</td>
<td></td>
</tr>
<tr>
<td>GNMT + RL [38]</td>
<td>24.6</td>
<td>39.92</td>
</tr>
<tr>
<td>ConvS2S [9]</td>
<td>25.16</td>
<td>40.46</td>
</tr>
<tr>
<td>MoE [32]</td>
<td>26.03</td>
<td>40.56</td>
</tr>
<tr>
<td>Deep-Att + PosUnk Ensemble [39]</td>
<td></td>
<td>40.4</td>
</tr>
<tr>
<td>GNMT + RL Ensemble [38]</td>
<td>26.30</td>
<td>41.16</td>
</tr>
<tr>
<td>ConvS2S Ensemble [9]</td>
<td>26.36</td>
<td>41.29</td>
</tr>
<tr>
<td>Transformer (base model)</td>
<td>27.3</td>
<td>38.1</td>
</tr>
<tr>
<td>Transformer (big)</td>
<td><strong>28.4</strong></td>
<td><strong>41.8</strong></td>
</tr>
</tbody>
</table>

- Big = 6 layers, 1000 dim for each token, 16 heads, base = 6 layers + other params halved

Vaswani et al. (2017)
Takeaways

- Can build MT systems with LSTM encoder-decoders, CNNs, or transformers
- Attention is very helpful for seq2seq models
- Also used for tasks including data-to-text generation and summarization
- Explicitly copying input can be beneficial as well
- Transformers are very strong models