CS378: Natural Language Processing
Lecture 19: MT 3, Transformers

Greg Durrett

The University of Texas at Austin
Announcements

- A3 back soon
- A4 due today
- A5 released today
- Final project released next week
Recall: Seq2seq Model

- Generate next word conditioned on previous word as well as hidden state

- \(W\) size is \(|\text{vocab}| \times |\text{hidden state}|\), softmax over entire vocabulary

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

\[
P(y|x) = \prod_{i=1}^{n} P(y_i|x, y_1, \ldots, y_{i-1})
\]

Decoder has separate parameters from encoder, so this can learn to be a language model (produce a plausible next word given current one)
Recall: Attention

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

- No attn: \( P(y_i|x, y_1, \ldots, y_{i-1}) = \text{softmax}(W\bar{h}_i) \)

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

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

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

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

- Weighted sum of input hidden states (vector)

- Some function \( f \)
Neural MT
Results: WMT English-French

- 12M sentence pairs

Classic PBMT system: \(~33\) BLEU, uses additional target-language data

- PBMT + rerank w/LSTMs: \(36.5\) BLEU (long line of work here; Devlin+ 2014)

Sutskever+ (2014) seq2seq single: \(30.6\) BLEU (input reversed)

Sutskever+ (2014) seq2seq ensemble: \(34.8\) BLEU

Luong+ (2015) seq2seq ensemble with attention and rare word handling: \(37.5\) BLEU

- But English-French is a really easy language pair and there’s *tons* of data for it! Does this approach work for anything harder?
Results: WMT English-German

- 4.5M sentence pairs

Classic phrase-based system: **20.7** BLEU

Luong+ (2014) seq2seq: **14** BLEU

Luong+ (2015) seq2seq ensemble with rare word handling: **23.0** BLEU

- Not nearly as good in absolute BLEU, but BLEU scores aren’t really comparable across languages

- French, Spanish = easiest
  German, Czech = harder

Japanese, Russian = hard (grammatically different, lots of morphology...)
**MT Examples**

<table>
<thead>
<tr>
<th>src</th>
<th>In einem Interview sagte Bloom jedoch, dass er und Kerr sich noch immer lieben.</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref</td>
<td>However, in an interview, Bloom has said that he and <em>Kerr</em> still love each other.</td>
</tr>
<tr>
<td>best</td>
<td>In an interview, however, Bloom said that he and <em>Kerr</em> still love.</td>
</tr>
<tr>
<td>base</td>
<td>However, in an interview, Bloom said that he and <em>Tina</em> were still <em>&lt;unk&gt;</em>.</td>
</tr>
</tbody>
</table>

- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn’t do this
### MT Examples

<table>
<thead>
<tr>
<th>src</th>
<th>Wegen der von Berlin und der Europäischen Zentralbank verhängten strengen Sparpolitik in Verbindung mit der Zwangsjacke, in die die jeweilige nationale Wirtschaft durch das Festhalten an der gemeinsamen Währung genötigt wird, sind viele Menschen der Ansicht, das Projekt Europa sei zu weit gegangen</th>
</tr>
</thead>
<tbody>
<tr>
<td>ref</td>
<td>The <em>austerity imposed by Berlin and the European Central Bank, coupled with the straitjacket</em> imposed on national economies through adherence to the common currency, has led many people to think Project Europe has gone too far.</td>
</tr>
<tr>
<td>best</td>
<td>Because of the strict <em>austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket</em> in which the respective national economy is forced to adhere to the common currency, many people believe that the European project has gone too far.</td>
</tr>
<tr>
<td>base</td>
<td>Because of the pressure <em>imposed by the European Central Bank and the Federal Central Bank with the strict austerity</em> imposed on the national economy in the face of the single currency, many people believe that the European project has gone too far.</td>
</tr>
</tbody>
</table>

- best = with attention, base = no attention

Luong et al. (2015)
Handling Rare Words

- Words are a difficult unit to work with: copying can be cumbersome, word vocabularies get very large

- Character-level models don’t work well

- Compromise solution: use thousands of “word pieces” (which may be full words but may also be parts of words)

  Input: _the _eco tax _port i co _in _Po nt - de - Bu is ...

  Output: _le _port ique _éco taxe _de _Pont - de - Bui s

- Can achieve transliteration with this, subword structure makes some translations easier to achieve

  Sennrich et al. (2016)
Byte Pair Encoding (BPE)

- Start with every individual byte (basically character) as its own symbol

```python
for i in range(num_merges):
    pairs = get_stats(vocab)
    best = max(pairs, key=pairs.get)
    vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences
- Merge the most frequent pair of adjacent characters

- Doing 8k merges => vocabulary of around 8000 word pieces. Includes many whole words

- Most SOTA NMT systems use this on both source + target

Sennrich et al. (2016)
**Byte Pair Encoding (BPE)**

<table>
<thead>
<tr>
<th>Original</th>
<th>BPE:</th>
<th>Unigram LM:</th>
<th>Original</th>
<th>BPE:</th>
<th>Unigram LM:</th>
</tr>
</thead>
<tbody>
<tr>
<td>furiously</td>
<td>.fur</td>
<td>ious</td>
<td>ly</td>
<td>tricycles</td>
<td>.t</td>
</tr>
<tr>
<td>Completely preposterous suggestions</td>
<td>.Comple</td>
<td>t</td>
<td>ely</td>
<td>.prep</td>
<td>ost</td>
</tr>
</tbody>
</table>

- BPE produces less linguistically plausible units than another technique based on a unigram language model.

Bostrom and Durrett (2020)
Google NMT
Google’s NMT System (2016)

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k

Wu et al. (2016)
Google’s NMT System (2016)

English-French:
Google’s phrase-based system: 37.0 BLEU
Luong+ (2015) seq2seq ensemble with rare word handling: 37.5 BLEU
Google’s 32k word pieces: 38.95 BLEU

English-German:
Google’s phrase-based system: 20.7 BLEU
Luong+ (2015) seq2seq ensemble with rare word handling: 23.0 BLEU
Google’s 32k word pieces: 24.2 BLEU

Wu et al. (2016)
Human Evaluation (En-Es)

- Similar to human-level performance on English-Spanish

Wu et al. (2016)
RNMT+ model — better RNNs in response to Transformers

<table>
<thead>
<tr>
<th>Model</th>
<th>RNMT+</th>
<th>Trans. Big</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>41.00</td>
<td>40.73</td>
</tr>
<tr>
<td>- Label Smoothing</td>
<td>40.33</td>
<td>40.49</td>
</tr>
<tr>
<td>- Multi-head Attention</td>
<td>40.44</td>
<td>39.83</td>
</tr>
<tr>
<td>- Layer Norm.</td>
<td>*</td>
<td>*</td>
</tr>
<tr>
<td>- Sync. Training</td>
<td>39.68</td>
<td>*</td>
</tr>
</tbody>
</table>

Table 4: Ablation results of RNMT+ and the Transformer Big model on WMT’14 En → Fr. We report average BLEU scores on the test set. An asterisk ‘*’ indicates an unstable training run (training halts due to non-finite elements).

RNMT+ is a bit better than Transformers, but also uses multi-head attention

Chen et al. (2018)
## Frontiers in MT: Small Data

<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>BLEU 100k</th>
<th>BLEU 3.2M</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>phrase-based SMT</td>
<td>15.87 ± 0.19</td>
<td>26.60 ± 0.00</td>
</tr>
<tr>
<td>2</td>
<td>NMT baseline</td>
<td>0.00 ± 0.00</td>
<td>25.70 ± 0.33</td>
</tr>
<tr>
<td>3</td>
<td>2 + ”mainstream improvements” (dropout, tied embeddings, layer normalization, bideep RNN, label smoothing)</td>
<td>7.20 ± 0.62</td>
<td>31.93 ± 0.05</td>
</tr>
<tr>
<td>4</td>
<td>3 + reduce BPE vocabulary (14k → 2k symbols)</td>
<td>12.10 ± 0.16</td>
<td>-</td>
</tr>
<tr>
<td>5</td>
<td>4 + reduce batch size (4k → 1k tokens)</td>
<td>12.40 ± 0.08</td>
<td>31.97 ± 0.26</td>
</tr>
<tr>
<td>6</td>
<td>5 + lexical model</td>
<td>13.03 ± 0.49</td>
<td>31.80 ± 0.22</td>
</tr>
<tr>
<td>7</td>
<td>5 + aggressive (word) dropout</td>
<td>15.87 ± 0.09</td>
<td><strong>33.60 ± 0.14</strong></td>
</tr>
<tr>
<td>8</td>
<td>7 + other hyperparameter tuning (learning rate, model depth, label smoothing rate)</td>
<td><strong>16.57 ± 0.26</strong></td>
<td>32.80 ± 0.08</td>
</tr>
<tr>
<td>9</td>
<td>8 + lexical model</td>
<td>16.10 ± 0.29</td>
<td>33.30 ± 0.08</td>
</tr>
</tbody>
</table>

- Synthetic small data setting: German -> English

Sennrich and Zhang (2019)
Particular interest in deploying MT systems for languages with little or no parallel data

BPE allows us to transfer models even without training on a specific language

Pre-trained models can help further

<table>
<thead>
<tr>
<th>Transfer</th>
<th>My→En</th>
<th>Id→En</th>
<th>Tr→En</th>
</tr>
</thead>
<tbody>
<tr>
<td>baseline (no transfer)</td>
<td>4.0</td>
<td>20.6</td>
<td>19.0</td>
</tr>
<tr>
<td>transfer, train</td>
<td>17.8</td>
<td>27.4</td>
<td>20.3</td>
</tr>
<tr>
<td>transfer, train, reset emb, train</td>
<td>13.3</td>
<td>25.0</td>
<td>20.0</td>
</tr>
<tr>
<td>transfer, train, reset inner, train</td>
<td>3.6</td>
<td>18.0</td>
<td>19.1</td>
</tr>
</tbody>
</table>

Table 3: Investigating the model’s capability to restore its quality if we reset the parameters. We use En→De as the parent.

Aji et al. (2020)
Transformers for MT
Self-attention Intro (notes)
Multi-Head Self Attention

- Multiple “heads” analogous to different convolutional filters
- Let $X = [\text{sent len, embedding dim}]$ be the input sentence
- Query $Q = W^Q X$: these are like the \textbf{decoder hidden state} in attention
- Keys $K = W^K X$: these control what gets attended to, along with the query
- Values $V = W^V X$: these vectors get summed up to form the output

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QQ^T}{\sqrt{d_k}})V$$

\[\text{dim of keys}\]

Vaswani et al. (2017)
# Multi-Head Self Attention

<table>
<thead>
<tr>
<th>Input</th>
<th>X₁</th>
<th>X₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedding</td>
<td><img src="image" alt="embedding" /></td>
<td><img src="image" alt="embedding" /></td>
</tr>
<tr>
<td>Queries</td>
<td>q₁</td>
<td>q₂</td>
</tr>
<tr>
<td>Keys</td>
<td>k₁</td>
<td>k₂</td>
</tr>
<tr>
<td>Values</td>
<td>v₁</td>
<td>v₂</td>
</tr>
</tbody>
</table>

Alammar, *The Illustrated Transformer*
Multi-Head Self Attention

Alammar, *The Illustrated Transformer*

\[
\text{sent len x sent len (attn for each word to each other)}
\]

\[
\begin{align*}
Q & = W^Q X \\
K & = W^K X \\
V & = W^V X \\
Z & = \text{softmax} \left( \frac{Q \cdot K^T}{\sqrt{d_k}} \right)
\end{align*}
\]

sent len x hidden dim

Z is a weighted combination of V rows
## Properties of Self-Attention

<table>
<thead>
<tr>
<th>Layer Type</th>
<th>Complexity per Layer</th>
<th>Sequential Operations</th>
<th>Maximum Path Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-Attention</td>
<td>( O(n^2 \cdot d) )</td>
<td>( O(1) )</td>
<td>( O(1) )</td>
</tr>
<tr>
<td>Recurrent</td>
<td>( O(n \cdot d^2) )</td>
<td>( O(n) )</td>
<td>( O(n) )</td>
</tr>
<tr>
<td>Convolutional</td>
<td>( O(k \cdot n \cdot d^2) )</td>
<td>( O(1) )</td>
<td>( O(\log_k(n)) )</td>
</tr>
<tr>
<td>Self-Attention (restricted)</td>
<td>( O(r \cdot n \cdot d) )</td>
<td>( O(1) )</td>
<td>( O(n/r) )</td>
</tr>
</tbody>
</table>

- \( n = \) sentence length, \( d = \) hidden dim, \( k = \) kernel size, \( r = \) restricted neighborhood size

- **Quadratic complexity**, but \( O(1) \) sequential operations (not linear like in RNNs) and \( O(1) \) “path” for words to inform each other

Vaswani et al. (2017)
Transformers

- Alternate multi-head self-attention layers and feedforward layers
- Residual connections let the model “skip” each layer — these are particularly useful for training deep networks

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

- If this is in a longer context, we want words to attend *locally*

- But transformers have *no notion of position* by default

Vaswani et al. (2017)
Transformers: Position Sensitivity

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)
Transformers

Alammar, The Illustrated Transformer
Transformers: Complete Model

- Encoder and decoder are both transformers.
- Decoder alternates attention over the output and attention over the input as well.
- Decoder consumes the previous generated tokens but has no recurrent state.

Vaswani et al. (2017)
### Transformers

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>EN-DE</td>
</tr>
<tr>
<td>ByteNet [18]</td>
<td>23.75</td>
</tr>
<tr>
<td>GNMT + RL [38]</td>
<td>24.6</td>
</tr>
<tr>
<td>ConvS2S [9]</td>
<td>25.16</td>
</tr>
<tr>
<td>MoE [32]</td>
<td>26.03</td>
</tr>
<tr>
<td>Deep-Att + PosUnk Ensemble [39]</td>
<td></td>
</tr>
<tr>
<td>GNMT + RL Ensemble [38]</td>
<td>26.30</td>
</tr>
<tr>
<td>ConvS2S Ensemble [9]</td>
<td>26.36</td>
</tr>
<tr>
<td>Transformer (base model)</td>
<td></td>
</tr>
<tr>
<td>Transformer (big)</td>
<td><strong>28.4</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)
Vaswani et al. (2017)
Takeaways

- Transformers are powerful seq2seq models, can also replace RNN encoders
- When you have massive datasets like for machine translation, transformers work very well
- Next: **pre-training** with transformer language models