CS388: Natural Language Processing

Lecture 17: Machine Translation 2

Greg Durrett

The University of Texas at Austin
Administrivia

- Project 2 due Thursday
Recall: Phrase-Based MT

Phrase table $P(f|e)$

Unlabeled English data

Language model $P(e)$

Noisy channel model:
combine scores from
translation model +
language model to
translate foreign to
English

“Translate faithfully but make fluent English”

$P(e|f) \propto P(f|e)P(e)$
Recall: HMM for Alignment

- Sequential dependence between a’s to capture monotonicity

\[ P(f, a | e) = \prod_{i=1}^{n} P(f_i | e_{a_i}) P(a_i | a_{i-1}) \]

\[ e \quad Thank \ you, \ I \ shall \ do \ so \ gladly. \]

\[ a \quad 0 \rightarrow 2 \rightarrow 6 \rightarrow 5 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 7 \rightarrow 8 \]

\[ f \quad Gracias, \ lo \ hare \ de \ muy \ buen \ grado. \]

- Alignment dist parameterized by jump size: \[ P(a_j - a_{j-1}) \]

- \[ P(f_i | e_{a_i}) \]: word translation table

Brown et al. (1993)
Recall: Decoding

Scores from language model $P(e) + \text{translation model } P(f|e)$
This Lecture

- Neural MT details
- Tokenization
- Google’s NMT system
- Transformers for MT
Neural MT
Encoder-Decoder MT

- Sutskever seq2seq paper: first major application of LSTMs to NLP
- Basic encoder-decoder with beam search

<table>
<thead>
<tr>
<th>Method</th>
<th>test BLEU score (ntst14)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single forward LSTM, beam size 12</td>
<td>26.17</td>
</tr>
<tr>
<td>Single reversed LSTM, beam size 12</td>
<td>30.59</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 1</td>
<td>33.00</td>
</tr>
<tr>
<td>Ensemble of 2 reversed LSTMs, beam size 12</td>
<td>33.27</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 2</td>
<td>34.50</td>
</tr>
<tr>
<td>Ensemble of 5 reversed LSTMs, beam size 12</td>
<td><strong>34.81</strong></td>
</tr>
</tbody>
</table>

- SOTA = 37.0 — not all that competitive...

Sutskever et al. (2014)
Better model from seq2seq lectures: encoder-decoder with attention and copying for rare words

distribution over vocab + copying

the movie was great
Results: WMT English-French

12M sentence pairs

Classic phrase-based system: \(\sim 33\) BLEU, uses additional target-language data

Rerank with LSTMs: \(36.5\) BLEU (long line of work here; Devlin+ 2014)

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

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
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

- BLEU isn’t comparable across languages, but this performance still isn’t as good
- French, Spanish = easiest
  - German, Czech, Chinese = 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 Kerr still love each other.</td>
</tr>
<tr>
<td>best</td>
<td>In an interview, however, Bloom said that he and Kerr still love.</td>
</tr>
<tr>
<td>base</td>
<td>However, in an interview, Bloom said that he and Tina were still &lt;unk&gt;.</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

Luong et al. (2015)
The austerity imposed by Berlin and the European Central Bank, coupled with the straitjacket imposed on national economies through adherence to the common currency, has led many people to think Project Europe has gone too far.

Because of the strict austerity measures imposed by Berlin and the European Central Bank in connection with the straitjacket 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.

Because of the pressure imposed by the European Central Bank and the Federal Central Bank with the strict austerity imposed on the national economy in the face of the single currency, many people believe that the European project has gone too far.

> best = with attention, base = no attention

Luong et al. (2015)
Backtranslation

- Classical MT methods used a bilingual corpus of sentences \(B = (S, T)\) and a large monolingual corpus \(T'\) to train a language model. Can neural MT do the same?

- Approach 1: force the system to generate \(T'\) as targets from null inputs
  
  \[
  s_1, t_1 \\
  s_2, t_2 \\
  ... \\
  [null], t'_1 \\
  [null], t'_2 \\
  ...
  \]

- Approach 2: generate synthetic sources with a T->S machine translation system (backtranslation)
  
  \[
  s_1, t_1 \\
  s_2, t_2 \\
  ... \\
  MT(t'_1), t'_1 \\
  MT(t'_2), t'_2 \\
  ...
  \]

Sennrich et al. (2015)
### Backtranslation

<table>
<thead>
<tr>
<th>name</th>
<th>training</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>tst2011</td>
</tr>
<tr>
<td>baseline (Gülçehre et al., 2015)</td>
<td></td>
<td>18.4</td>
</tr>
<tr>
<td>deep fusion (Gülçehre et al., 2015)</td>
<td></td>
<td>20.2</td>
</tr>
<tr>
<td>baseline parallel</td>
<td>parallel</td>
<td>18.6</td>
</tr>
<tr>
<td>parallel synth</td>
<td>parallel/parallel synth</td>
<td>19.9</td>
</tr>
<tr>
<td>Gigaword mono</td>
<td>parallel/Gigaword mono</td>
<td>18.8</td>
</tr>
<tr>
<td>Gigaword synth</td>
<td>parallel/Gigaword synth</td>
<td><strong>21.2</strong></td>
</tr>
</tbody>
</table>

- **Gigaword**: large monolingual English corpus
- **parallel synth**: backtranslate training data; makes additional noisy source sentences which could be useful

Sennrich et al. (2015)
Tokenization
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 _Pont - de - Bu is:

Output: _le _port ique _éco taxe _de _Pont - de - Buis:

- 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 in dictionary
- Merge the most frequent pair of adjacent characters

- Vocabulary stats are weighted over a large corpus

- Doing 30k merges => vocabulary of around 30,000 word pieces. Includes many whole words
  
  and there were no re_fueling stations anywhere
  
  one of the city’s more un_princi_pled real estate agents

Sennrich et al. (2016)
Word Pieces

- Alternative to BPE

while voc size < target voc size:
  - Build a language model over your corpus
  - Merge pieces that lead to highest improvement in language model perplexity

- Issues: what LM to use? How to make this tractable?
- SentencePiece library from Google: unigram LM
- Result: way of segmenting input appropriate for translation

Schuster and Nakajima (2012), Wu et al. (2016), Kudo and Richardson (2018)
Comparison

<table>
<thead>
<tr>
<th>Original:</th>
<th>furiously</th>
<th>Original:</th>
<th>tricycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a) BPE:</td>
<td>_fur</td>
<td>(b) BPE:</td>
<td>_t</td>
</tr>
<tr>
<td></td>
<td>iously</td>
<td></td>
<td>ric</td>
</tr>
<tr>
<td>Unigram LM:</td>
<td>_fur</td>
<td>Unigram LM:</td>
<td>_tri</td>
</tr>
<tr>
<td></td>
<td>ious</td>
<td></td>
<td>cycle</td>
</tr>
<tr>
<td></td>
<td>ly</td>
<td></td>
<td>s</td>
</tr>
<tr>
<td>(c) BPE:</td>
<td>_Complet</td>
<td></td>
<td>_suggest</td>
</tr>
<tr>
<td></td>
<td>t</td>
<td></td>
<td>ions</td>
</tr>
<tr>
<td>Unigram LM:</td>
<td>_Complete</td>
<td></td>
<td>_suggestion</td>
</tr>
<tr>
<td></td>
<td>ly</td>
<td></td>
<td>s</td>
</tr>
<tr>
<td></td>
<td>_pre</td>
<td></td>
<td>_post</td>
</tr>
<tr>
<td></td>
<td>_er</td>
<td></td>
<td>_ous</td>
</tr>
</tbody>
</table>

- BPE produces less linguistically plausible units than word pieces (unigram LM)
- Some evidence that unigram LM works better in pre-trained transformer models

Bostrom and Durrett (2020)
Subword Regularization

<table>
<thead>
<tr>
<th>Subwords (. means spaces)</th>
<th>Vocabulary id sequence</th>
</tr>
</thead>
<tbody>
<tr>
<td>.Hell/o/.world</td>
<td>13586 137 255</td>
</tr>
<tr>
<td>.H/ello/.world</td>
<td>320 7363 255</td>
</tr>
<tr>
<td>.He/llo/.world</td>
<td>579 10115 255</td>
</tr>
<tr>
<td>/.He/l/l/o/.world</td>
<td>7 18085 356 356 137 255</td>
</tr>
<tr>
<td>.H/el/l/o/.world</td>
<td>320 585 356 137 7 12295</td>
</tr>
</tbody>
</table>

- Change subword sampling on-the-fly during training

<table>
<thead>
<tr>
<th>Domain (size)</th>
<th>Corpus</th>
<th>Language pair</th>
<th>Baseline (BPE)</th>
<th>Proposed (SR)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web (5k)</td>
<td>IWSLT15</td>
<td>en → vi</td>
<td>13.86</td>
<td>17.36*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vi → en</td>
<td>7.83</td>
<td>11.69*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>en → zh</td>
<td>9.71</td>
<td>13.85*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>zh → en</td>
<td>5.93</td>
<td>8.13*</td>
</tr>
<tr>
<td></td>
<td>IWSLT17</td>
<td>en → fr</td>
<td>16.09</td>
<td>20.04*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>fr → en</td>
<td>14.77</td>
<td>19.99*</td>
</tr>
<tr>
<td></td>
<td>WMT14</td>
<td>en → de</td>
<td>22.71</td>
<td>26.02*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>de → en</td>
<td>26.42</td>
<td>29.63*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>en → cs</td>
<td>19.53</td>
<td>21.41*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>cs → en</td>
<td>25.94</td>
<td>27.86*</td>
</tr>
</tbody>
</table>

- Subword regularization (SR) improves results over a static scheme (BPE)

Kudo (2018)
Google NMT
Google’s NMT System

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

Wu et al. (2016)
Google’s NMT System

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)
## Google’s NMT System

<table>
<thead>
<tr>
<th>Source</th>
<th>English</th>
<th>French</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>She was spotted three days later by a dog walker trapped in the quarry</td>
<td>Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière</td>
<td>6.0</td>
</tr>
<tr>
<td>PBMT</td>
<td>Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière</td>
<td>Elle a été repéré trois jours plus tard par un promeneur de chien piégé dans la carrière</td>
<td>2.0</td>
</tr>
<tr>
<td>GNMT</td>
<td>Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière</td>
<td>Elle a été repérée trois jours plus tard par un traîneau à chiens piégé dans la carrière</td>
<td>2.0</td>
</tr>
<tr>
<td>Human</td>
<td>Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière</td>
<td>Elle a été repérée trois jours plus tard par une personne qui promenait son chien coincée dans la carrière</td>
<td>5.0</td>
</tr>
</tbody>
</table>

Gender is correct in GNMT but not in PBMT

"sled" and "walker"
## 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
Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word
  \[ \alpha_{i,j} = \text{softmax}(x_i^\top x_j) \quad \text{scalar} \]
  \[ x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \quad \text{vector} = \text{sum of scalar} \times \text{vector} \]

- Multi-head self attention: we are going to replicate this machinery several times with different parameters

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

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

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

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

Alammar, The Illustrated Transformer
Multi-Head Self Attention

Alammar, *The Illustrated Transformer*

sent len x sent len (attn for each word to each other)

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

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
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
### 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>27.3</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)
Visualiza\textsuperscript{1}tion

Visualiza\textsuperscript{1}tion

Vaswani et al. (2017)
Takeaways

- Can build MT systems with LSTM encoder-decoders or transformers (or CNNs)
- Word piece / byte pair models are really effective and easy to use
- State of the art systems are getting pretty good, but lots of challenges remain, especially for low-resource settings
- Next time: pre-trained transformer models (BERT), applied to other tasks