Attention

- Encoder hidden states capture contextual source word identity
- Decoder hidden states are now mostly responsible for selecting what to attend to
- Doesn’t take a complex hidden state to walk monotonically through a sentence and spit out word-by-word translations

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

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

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

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 _eco taxe _de _Pont - de - Buis..._.

- Can achieve transliteration with this, subword structure makes some translations easier to achieve
### Byte Pair Encoding (BPE)

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

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

Gender is correct in GNMT but not in PBMT

“sled” “walker”

Wu et al. (2016)

QA Intro

Combinatory Categorial Grammar

‣ Steedman+Szabolcsi (1980s): formalism bridging syntax and semantics
‣ Parallel derivations of syntactic parse and lambda calculus expression
‣ Syntactic categories (for this lecture): S, NP, “slash” categories
‣ S\NP: “If I combine with an NP on my left side, I form a sentence” — verb
‣ When you apply this, there has to be a parallel instance of function application on the semantics side

Combinatory Categorial Grammar
Combinatory Categorial Grammar

- Steedman+Szabolcsi 1980s: formalism bridging syntax and semantics
- Syntactic categories (for this lecture): S, NP, “slash” categories
- S\NP: “if I combine with an NP on my left side, I form a sentence” — verb
- (S\NP)/NP: “I need an NP on my right and then on my left” — verb with a direct object

Ordering:
- S
  - \(\text{sings(e728)}\)
- S\NP
  - \(\text{\lambda y.\; \text{sings(y)}}\)
- NP
  - e728
  - \(\text{\lambda y.\; \text{sings(y)}}\)
- S\NP/NP
  - \(\text{\lambda x.\; \text{\lambda y.\; borders(y, e89)}}\)

Example sentence:
- Eminem sings (\(\text{sings(e728)}\))
- Oklahoma borders (\(\text{\lambda y.\; \text{\lambda x.\; borders(y, x)}}\))
- Texas

CCG Parsing

\[
\begin{align*}
\frac{(S/(S/NP))/N}{\lambda f.\lambda g.\lambda x.\; f(x) \land g(x)} & \quad \frac{N}{\lambda x.\; \text{state}(x)} & \quad \frac{(S\setminus NP)/NP}{\lambda x.\; \text{\lambda y.\; borders(y, x)}} & \quad \frac{NP}{\text{\lambda y.\; \text{\lambda x.\; borders(y, tex)}}} \\
\frac{\text{states}}{\lambda x.\; \text{\lambda y.\; borders(y, x)}} & \quad \frac{\text{Texas}}{\text{\lambda y.\; \text{\lambda x.\; borders(y, tex)}}} \\
\frac{\text{What}}{\lambda x.\; \text{\lambda y.\; borders(y, tex)}} & \quad \frac{\text{What}}{\lambda x.\; \text{\lambda y.\; borders(y, tex)}} & \quad \frac{\text{What}}{\lambda x.\; \text{\lambda y.\; borders(y, tex)}} & \quad \frac{\text{What}}{\lambda x.\; \text{\lambda y.\; borders(y, tex)}}
\end{align*}
\]

- "What" is a very complex type: needs a noun and needs a S\NP to form a sentence. S\NP is basically a verb phrase (\text{\lambda x.\; \text{\lambda y.\; borders(y, tex)}})

Zettlemoyer and Collins (2005)

CCG Parsing

- These question are compositional: we can build bigger ones out of smaller pieces
  - \text{What states border Texas?}
  - \text{What states border states bordering Texas?}
  - \text{What states border states bordering states bordering Texas?}

- In general, answering this does require parsing and not just slot-filling

Zettlemoyer and Collins (2005)
Training CCG Parsers

- Training data looks like pairs of sentences and logical forms
- Unlike PCFGs, we don’t know which words yielded which fragments of CCG
- Requires an “unsupervised” approach like Model 1 for word alignment

What states border Texas
\[ \lambda x. \text{state}(x) \land \text{borders}(x, e89) \]

What borders Texas
\[ \lambda x. \text{borders}(x, e89) \]

Zettlemoyer and Collins (2005)

Seq2seq Semantic Parsing

Semantic Parsing as Translation

“what states border Texas”
\[ \lambda x. (\text{state}(x) \land \text{borders}(x, e89)) \]

Write down a linearized form of the semantic parse, train seq2seq models to directly translate into this representation

What are some benefits of this approach compared to grammar-based?

Handing Invariances

“what states border Texas”
“what states border Ohio”

- Parsing-based approaches handle these the same way
- Possible divergences: features, different weights in the lexicon
- Can we get seq2seq semantic parsers to handle these the same way?
- Key idea: do data augmentation by synthetically creating more data from a single example

Jia and Liang (2016)
Semantic Parsing as Translation

**Prolog**

```prolog
x: "what is the population of iowa?" y: _answer ( NV , ( _population ( NV , VI ) , _const ( VO , _stateid ( iowa )) ) )
```

**Lambda calculus**

```lambda
x: "can you list all flights from chicago to milwaukee" y: \(\lambda\) ( _lambda x0 e ( \(\lambda\) and ( x0 (chicago ) ) ( __to __x milwaukee __x )))
```

**Other DSLs**

- Handle all of these with uniform machinery! Jia and Liang (2016)

Three forms of data augmentation all help

Results on these tasks are still not as strong as hand-tuned systems from 10 years ago, but the same simple model can do well at all problems

<table>
<thead>
<tr>
<th>Previous Work</th>
<th>Geo</th>
<th>ATIS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zettlemoyer and Collins (2007)</td>
<td>88.9</td>
<td>91.0</td>
</tr>
<tr>
<td>Kwiatkowski et al. (2010)</td>
<td>91.0</td>
<td>100.0</td>
</tr>
<tr>
<td>Liang et al. (2011)</td>
<td>89.6</td>
<td>82.8</td>
</tr>
<tr>
<td>Kwiatkowski et al. (2011)</td>
<td>86.5</td>
<td>83.5</td>
</tr>
<tr>
<td>Poon (2013)</td>
<td>88.9</td>
<td>84.2</td>
</tr>
<tr>
<td>Zhao and Huang (2015)</td>
<td>88.9</td>
<td>84.2</td>
</tr>
</tbody>
</table>

Our Model

- No Recombination: 85.0 76.3
- ABSENTITIES: 85.4 79.9
- ABSWHOLEPHRASES: 87.5 81.3
- CONCAT-2: 84.6 79.0
- CONCAT-3: 77.5
- AWP + AE: 88.9 78.8
- AE + C2: 89.3 83.3

Applications

- GeoQuery (Zelle and Mooney, 1996): answering questions about states (~80% accuracy)
- Jobs: answering questions about job postings (~80% accuracy)
- ATIS: flight search
- Can do well on all of these tasks if you handcraft systems and use plenty of training data: these domains aren’t that rich

Next Time

- QA from raw text: how do we answer a question about a passage?
- Neural networks for QA
- Final project discussion