Recall:

**Phrase-Based MT**
- Phrase table $P(f|e)$
- Language model $P(e)$
- Unlabeled English data

```
<table>
<thead>
<tr>
<th>English</th>
<th>French</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>cat</td>
<td>chat</td>
<td>0.9</td>
</tr>
<tr>
<td>dog</td>
<td>chien</td>
<td>0.8</td>
</tr>
<tr>
<td>house</td>
<td>maison</td>
<td>0.6</td>
</tr>
<tr>
<td>my house</td>
<td>ma maison</td>
<td>0.9</td>
</tr>
<tr>
<td>language</td>
<td>langue</td>
<td>0.9</td>
</tr>
</tbody>
</table>
```

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

- “Translate faithfully but make fluent English”

**Recall: HMM for Alignment**
- Sequential dependence between a’s to capture monotonicity

$$P(f, a|e) = \prod_{i=1}^{n} P(f_{i}|e_{a_{i}})P(a_{i}|a_{i-1})$$

- $e$: Thank you, I shall do so gladly.
- $a$: 0 2 6 5 7 7 7 8
- $f$: 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 $P(e)$ + translation model $P(f|e)$

This Lecture

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

- Better model from seq2seq lectures: encoder-decoder with attention and copying for rare words
- distribution over vocab + copying

Results: WMT English-French

12M sentence pairs
- Classic phrase-based system: ~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…)
- best = with attention, base = no attention
- NMT systems can hallucinate words, especially when not using attention — phrase-based doesn’t do this

MT Examples

| src  | In einem Interview sagte Bloom jedoch, dass er und Kerr sich noch immer lieben. |
| ref  | However, in an interview, Bloom has said that he and Kerr still love each other. |
| best | In an interview, however, Bloom said that he and Kerr still love. |
| base | However, in an interview, Bloom said that he and Tina were still <unk>. |

Luong et al. (2015)
MT Examples

Luong et al. (2015)

| src | 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. |
| ref | 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. |
| best | 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. |
| base | 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

Backtranslation

Sennrich et al. (2015)

‣ 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

\[
\begin{align*}
& s_1, t_1 \\
& s_2, t_2 \\
& \ldots \\
& \{null\}, t'_1 \\
& \{null\}, t'_2 \\
& \ldots
\end{align*}
\]

‣ Approach 2: generate synthetic sources with a $T\rightarrow S$ machine translation system (backtranslation)

\[
\begin{align*}
& s_1, t_1 \\
& s_2, t_2 \\
& \ldots \\
& MT(t'_1), t'_1 \\
& MT(t'_2), t'_2 \\
& \ldots
\end{align*}
\]

Sennrich et al. (2015)

Backtranslation

<table>
<thead>
<tr>
<th>name</th>
<th>training data</th>
<th>instances</th>
<th>BLEU</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>tst2011</td>
<td>tst2012</td>
</tr>
<tr>
<td>baseline (Güçlüehre et al., 2015)</td>
<td></td>
<td></td>
<td>18.4</td>
<td>18.8</td>
</tr>
<tr>
<td>deep fusion (Güçlüehre et al., 2015)</td>
<td></td>
<td></td>
<td>20.2</td>
<td>20.2</td>
</tr>
<tr>
<td>baseline</td>
<td>parallel</td>
<td>7.2m</td>
<td>18.6</td>
<td>18.2</td>
</tr>
<tr>
<td>parallel</td>
<td>parallel/parallel</td>
<td>6m/6m</td>
<td>19.9</td>
<td>20.4</td>
</tr>
<tr>
<td>Gigawordsmono</td>
<td>parallel/Gigawordsmono</td>
<td>7.6m/7.6m</td>
<td>18.8</td>
<td>19.6</td>
</tr>
<tr>
<td>Gigawordsynth</td>
<td>parallel/Gigawordsynth</td>
<td>8.4m/8.4m</td>
<td><strong>21.2</strong></td>
<td><strong>21.1</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_Po nt - de - Bu is...

  Output: _le_port ique_eco 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

  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

- 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

- Change subword sampling on-the-fly during training

Subword regularisation (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)

Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

Google NMT

- Google’s NMT System

- 8-layer LSTM encoder-decoder with attention, word piece vocabulary of 8k-32k  
Wu et al. (2016)

- Change subword sampling on-the-fly during training

Subword regularization (SR) improves results over a static scheme (BPE)  
Kudo (2018)
Human Evaluation (En-Es)

- Similar to human-level performance on English-Spanish

Google’s NMT System

- Gender is correct in GNMT but not in PBMT
- “sled” “walker”

Frontiers in MT: Small Data

- Synthetic small data setting: German -> English

Frontiers in MT: Low-Resource

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

<table>
<thead>
<tr>
<th>Source</th>
<th>System</th>
<th>Translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>PBMT</td>
<td>Elle a été repérée trois jours plus tard par un promeneur de chiens piégé dans la carrière.</td>
<td>6.0</td>
</tr>
<tr>
<td>GNMT</td>
<td>Elle a été repérée trois jours plus tard par un promeneur de 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>5.0</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.

Wu et al. (2016)

Sennrich and Zhang (2019)

Aji et al. (2020)
Transformers for MT

Recall: Self-Attention

- Each word forms a “query” which then computes attention over each word
  \[ \alpha_{i,j} = \text{softmax}(x_i^T x_j) \text{ scalar} \]
  \[ x'_i = \sum_{j=1}^{n} \alpha_{i,j} x_j \text{ vector = sum of scalar * vector} \]

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

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 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

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

dim of keys

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

- **Alammar, The Illustrated Transformer**
  - sent len x sent len (attn for each word to each other)
  - softmax\(\left(\begin{array}{c}
  Q \\
  K^T
  \end{array}\right)\)
  - 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_2(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

---

**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

---

**Transformers: Position Sensitivity**

- 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.

- 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.

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