Armed with the idea of language models ($P(w)$) and Transformers (good models for this), we still need to actually put together an MT system.

Sequence-to-sequence (seq2seq) models: we define these as distributions $P(y|x)$ and decide how to train and do inference. Training looks like LM training, inference is new.

Subword tokenization: key practical implementation detail

Vaswani et al. (2017)
**Transformers: Complete Model**

- Transformer encoder (A4) + decoder (looks back at encoder, but similar architecture)
- Decoder alternates attention over the output and attention over the input as well
- Decoder **consumes the previous generated tokens**. You need to run the whole decoder to predict token 1 of the output, then run the decoder again to predict token 2, etc.

Vaswani et al. (2017)

**Seq2seq Model**

- Generate next word conditioned on previous words (like a language model) and conditioned on the source
- W size is |vocab| x |hidden state|, softmax over entire vocabulary

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

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

Decoder learns to both condition on \( \mathbf{x} \) and generate plausible sequences \( \mathbf{y} \) (if trained well/on enough data)

**Inference (“Decoding”)**

- During inference: need to compute the argmax over the word predictions and then run the next step of the decoder (which looks back at all previous encoder + decoder steps)
- Need to actually evaluate computation graph up to this point
- Decoder is advanced one state at a time until [STOP] is reached

**Training**

- Objective: maximize \[
\sum_{(\mathbf{x}, \mathbf{y})} \sum_{i=1}^{n} \log P(y_i^*|\mathbf{x}, y_1^*, \ldots, y_{i-1}^*)
\]
- One loss term for each target-sentence word, feed the correct word regardless of model’s prediction (called “teacher forcing”). **Can train in “one go” like the language model, no need to run each step sequentially.**
Training: Scheduled Sampling

- Model needs to do the right thing even with its own predictions

- Scheduled sampling: with probability $p$, take the gold as input, else take the model’s prediction
- Starting with $p = 1$ (teacher forcing) and decaying it works best
- Not really used these days

Decoding Methods

Decoding Strategies

- LMs place a distribution $P(y_i \mid y_1, \ldots, y_{i-1})$
- seq2seq models place a distribution $P(y_i \mid x, y_1, \ldots, y_{i-1})$

- Generation from both models looks similar; how do we do it?
  - Option 1: max $y_i P(y_i \mid y_1, \ldots, y_{i-1})$ — take greedily best option
  - Option 2: use beam search to find the sequence with the highest prob.
  - Option 3: sample from the model; draw $y_i$ from that distribution
  - Machine translation: use beam search. The top-scoring hypothesis is usually a great translation

Story generation (this is with GPT-2):

Context: In a shocking finding, scientists discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.

Beam Search, $b=32$:
"The study, published in the Proceedings of the National Academy of Sciences of the United States of America (PNAS), was conducted by researchers from the Universidad Nacional Autónoma de México (UNAM) and the Universidad Nacional Autónoma de México (UNAM/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de México/Universidad Nacional Autónoma de ..."

Pure Sampling:
They were cattle called Bolivian Cavalleros; they live in a remote desert uninterrupted by town, and they speak huge, beautiful, paradisical Bolivian linguistic thing. They say, ‘Lunch, marge’ They don’t tell what the lunch is.” director Professor Chipers Omwell told Sky News. “They’ve only been talking to scientists, like we’re being interviewed by TV reporters. We don’t even stick around to be interviewed by TV reporters. Maybe that’s how they figured out that they’re cosplaying as the Bolivian Cavalleros.”

- Beam search degenerates and starts repeating. If you see a fragment repeated 2-3x, it has very high probability to keep repeating
- Sampling is too noisy — introduces many grammatical errors

Holtzman et al. (2019)
Degeneration

- Beam search fails because the model is *locally normalized*
- Let’s look at all the individual decisions that get made here

\[
P(\text{Nacional} \mid \ldots \text{Universidad}) \text{ is high} \\
P(\text{Autónoma} \mid \ldots \text{Universidad Nacional}) \text{ is high} \\
P(\text{México} \mid \text{Universidad Nacional Autónoma de}) \text{ is high} \\
\]

- Each word is likely given the previous words but the sequence is bad

Holtzman et al. (2019)

Drawbacks of Sampling

- Sampling is “too random”

\[
P(y \mid \ldots \text{they live in a remote desert uninterrupted by}) \\
0.01 \text{ roads} \\
0.01 \text{ towns} \\
0.01 \text{ people} \\
0.005 \text{ civilization} \\
\ldots \\
0.0005 \text{ town} \\
\]

Good options, maybe accounting for 90% of the total probability mass. So a 90% chance of getting something good

Long tail with 10% of the mass

Holtzman et al. (2019)

Nucleus Sampling

\[
P(y \mid \ldots \text{they live in a remote desert uninterrupted by}) \\
0.01 \text{ roads} \\
0.01 \text{ towns} \\
0.01 \text{ people} \\
0.005 \text{ civilization} \\
\]

renormalize and sample

\[
\text{cut off after } p\% \text{ of mass} \\
\]

- Define a threshold \( p \). Keep the most probable options account for \( p\% \) of the probability mass (the nucleus), then sample among these.
- To implement: sort options by probability, truncate the list once the total exceeds \( p \), then renormalize and sample from it

Holtzman et al. (2019)

Decoding Strategies

- LMs place a distribution \( P(y_i \mid y_1, \ldots, y_{i-1}) \)
- seq2seq models place a distribution \( P(y_i \mid x, y_1, \ldots, y_{i-1}) \)

- How to generate sequences?
  - Option 1: \( \max y_i P(y_i \mid y_1, \ldots, y_{i-1}) \) — take greedily best option
  - Option 2: use beam search to find the sequence with the highest prob.
  - Option 3: sample from the model, draw \( y \) from that distribution
  - Option 4: nucleus sampling

Holtzman et al. (2019)
Handling Rare Words

- Words are a difficult unit to work with: copying can be cumbersome, word vocabularies get very large
- When you have 100,000+ words, the final matrix multiply and softmax start to dominate the computation
- Character-level models were explored extensively in 2016-2018 but simply don’t work well — becomes very expensive to represent sequences

Subword Tokenization

- Subword tokenization: wide range of schemes that use tokens that are between characters and words in terms of granularity
- These “word pieces” may be full words or parts of words
  - Input: _the _eco tax _port i co _in _Po nt - de - Bu is ...
  - _ indicates the word piece starting a word (can think of it as the space character).

Sennrich et al. (2016)

Subword Tokenization

- Subword tokenization: wide range of schemes that use tokens that are between characters and words in terms of granularity
- These “word pieces” may be full words or 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>furiously</th>
<th>BPE:</th>
<th>tricycles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram LM:</td>
<td>.fur</td>
<td>iously</td>
<td>.t</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Original:</th>
<th>Completely preposterous suggestions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unigram LM:</td>
<td>.Complete</td>
</tr>
</tbody>
</table>

- BPE produces less linguistically plausible units than another technique based on a unigram language model: rather than greedily merge, find chunks which make the sequence look likely under a unigram LM

Bostrom and Durrett (2020)

---

**Tokenization Today**

- **All pre-trained** models use some kind of subword tokenization with a tuned vocabulary; usually between 50k and 250k pieces (larger number of pieces for multilingual models)

- As a result, classical word embeddings like GloVe are not used. All subword representations are randomly initialized and learned in the Transformer models

---

**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

| 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>. |

- 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)

MT Examples

| 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

Luong et al. (2015)
Google NMT (2016)

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)

- Similar to human-level performance on English-Spanish

Wu et al. (2016)

Human Evaluation (En-Es)

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)
Transformer MT + Frontiers

Transformers

<table>
<thead>
<tr>
<th>Model</th>
<th>BLEU</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ByteNet [18]</td>
<td>23.75</td>
<td>EN-DE</td>
<td>39.2</td>
</tr>
<tr>
<td>GNMT + RL [38]</td>
<td>25.16</td>
<td>40.46</td>
<td></td>
</tr>
<tr>
<td>ConvS2S [9]</td>
<td>26.03</td>
<td>40.56</td>
<td></td>
</tr>
<tr>
<td>GNMT + RL Ensemble [38]</td>
<td>26.36</td>
<td>41.29</td>
<td></td>
</tr>
<tr>
<td>ConvS2S Ensemble [9]</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

Vaswani et al. (2017)

Frontiers in MT: Small Data

<table>
<thead>
<tr>
<th>ID</th>
<th>system</th>
<th>BLEU</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>phrase-based SMT</td>
<td>15.87</td>
<td>26.60 ± 0.00</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>NMT baseline</td>
<td>0.00</td>
<td>25.70 ± 0.33</td>
<td></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</td>
<td>31.93 ± 0.05</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>3 + reduce BPE vocabulary (14k → 2k symbols)</td>
<td>12.10</td>
<td>12.40 ± 0.08</td>
<td>31.97 ± 0.26</td>
</tr>
<tr>
<td>5</td>
<td>4 + reduce batch size (4k → 1k tokens)</td>
<td>13.03</td>
<td>13.03 ± 0.49</td>
<td>31.80 ± 0.22</td>
</tr>
<tr>
<td>6</td>
<td>5 + lexical model</td>
<td></td>
<td>16.57 ± 0.26</td>
<td>32.80 ± 0.08</td>
</tr>
<tr>
<td>7</td>
<td>5 + aggressive (word) dropout</td>
<td></td>
<td>16.10 ± 0.29</td>
<td>33.30 ± 0.08</td>
</tr>
<tr>
<td>8</td>
<td>7 + other hyperparameter tuning (learning rate, model depth, label smoothing rate)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>8 + lexical model</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

‣ Synthetic small data setting: German -> English

Sennrich and Zhang (2019)

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

Burmese, Indonesian, Turkish

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

Aji et al. (2020)
Frontiers in MT: Multilingual Models

Yinhan Liu et al. (2020)

Transformers are state-of-the-art for machine translation

They work really well on languages where we have a ton of data. When they don’t: pre-training can help

Next up: exploring pre-training in more detail (ELMo, BERT, GPT, etc.)

Takeaways

Yinhan Liu et al. (2020)

Multilingual Denoising Pre-Training (mBART)

Fine-tuning on Machine Translation

Random = random initialization

Yinhan Liu et al. (2020)