L17: Neural Machine Translation - 2

Spring 2021
• **Encode** entire input sequence into a single vector *(using an RNN)*

• **Decode** one word at a time *(again, using an RNN!)*

*Sutskever et al., 2014*
How seq2seq changed the MT landscape
MT Progress

(source: Rico Sennrich)
Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation

Table 10: Mean of side-by-side scores on production data

<table>
<thead>
<tr>
<th></th>
<th>PBMT</th>
<th>GNMT</th>
<th>Human</th>
<th>Relative Improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>English → Spanish</td>
<td>4.885</td>
<td>5.428</td>
<td>5.504</td>
<td>87%</td>
</tr>
<tr>
<td>English → French</td>
<td>4.932</td>
<td>5.295</td>
<td>5.496</td>
<td>64%</td>
</tr>
<tr>
<td>English → Chinese</td>
<td>4.035</td>
<td>4.594</td>
<td>4.987</td>
<td>58%</td>
</tr>
<tr>
<td>Spanish → English</td>
<td>4.872</td>
<td>5.187</td>
<td>5.372</td>
<td>63%</td>
</tr>
<tr>
<td>French → English</td>
<td>5.046</td>
<td>5.343</td>
<td>5.404</td>
<td>83%</td>
</tr>
<tr>
<td>Chinese → English</td>
<td>3.694</td>
<td>4.263</td>
<td>4.636</td>
<td>60%</td>
</tr>
</tbody>
</table>

(Wu et al., 2016)
Versatile seq2seq

- Seq2seq finds applications in many other tasks!
- Any task where inputs and outputs are sequences of words/characters
  - Summarization (input text → summary)
  - Dialogue (previous utterance → reply)
  - Parsing (sentence → parse tree in sequence form)
  - Question answering (context+question → answer)
Issues with vanilla seq2seq

- A single encoding vector, $h^{enc}$, needs to capture all the information about source sentence
- Longer sequences can lead to vanishing gradients
- Model may “overfit” to training sequences
Issues with vanilla seq2seq

- A single encoding vector, $h^{enc}$, needs to capture all the information about source sentence
- Longer sequences can lead to vanishing gradients
- Model may “overfit” to training sequences
Remember alignments?

\[
ap = (3, 4, 2, 1)^\top
\]

\[
ap = (1, 2, 3, 0, 4)^\top
\]
The neural MT equivalent of alignment models

**Key idea:** At each time step during decoding, **focus on a particular part** of source sentence

- This depends on the decoder’s current hidden state $h^{dec}$ (i.e. an idea of what you are trying to decode)

- Usually implemented as a probability distribution over the hidden states of the encoder ($h^{enc}$)
Seq2seq with attention

(Ask credit: Abigail See)
On this decoder timestep, we’re mostly focusing on the first encoder hidden state ("he")

Take softmax to turn the scores into a probability distribution

Source sentence (input)
Use the attention distribution to take a **weighted sum** of the encoder hidden states.

The attention output mostly contains information from the **hidden states** that received high attention.
Concatenate attention output with decoder hidden state, then use to compute $\hat{y}_1$ as before.
Computing attention

- Encoder hidden states: $h_1^{enc}, \ldots, h_n^{enc}$
- Decoder hidden state at time $t$: $h_t^{dec}$
- First, get attention scores for this time step of decoder (we’ll define $g$ soon):
  \[ e^t = [g(h_1^{enc}, h_t^{dec}), \ldots, g(h_n^{enc}, h_t^{dec})] \]
- Obtain the attention distribution using softmax:
  \[ \alpha^t = \text{softmax}(e^t) \in \mathbb{R}^n \]
- Compute weighted sum of encoder hidden states:
  \[ a_t = \sum_{i=1}^{n} \alpha_i^t h_i^{enc} \in \mathbb{R}^h \]
- Finally, concatenate with decoder state and pass on to output layer:
  \[ [a_t; h_t^{dec}] \in \mathbb{R}^{2h} \]
Je suis étudiant

Encoder hidden state

hidden state #1

hidden state #2

hidden state #3
Types of attention

- Assume encoder hidden states $h_1^{enc}, h_2^{enc}, \ldots, h_n^{enc}$ and a decoder hidden state $h^{dec}$

1. **Dot-product attention** (assumes equal dimensions for $h^{enc}$ and $h^{dec}$):
   
   $$ e_i = g(h_i^{enc}, h^{dec}) = (h^{dec})^T h_i^{enc} \in \mathbb{R} $$

2. **Multiplicative attention**:
   
   $$ g(h_i^{enc}, h^{dec}) = (h^{dec})^T W h_i^{enc} \in \mathbb{R}, \text{ where } W \text{ is a weight matrix (learned)} $$

3. **Additive attention**:
   
   $$ g(h_i^{enc}, h^{dec}) = v^T \tanh (W_1 h_i^{enc} + W_2 h^{dec}) \in \mathbb{R} $$
   
   where $W_1, W_2$ are weight matrices (learned) and $v$ is a weight vector (learned)
Dot-product attention:
\[ g(h_i^{enc}, h^{dec}) = h^{dec} \cdot h^{enc} \]

Assuming we use dot product attention, which input word will have the highest attention value at current time step?

A) the  
B) cat  
C) sat
What if we use multiplicative attention with $W = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$?

Which input word will have the highest attention value at current time step?

A) the
B) cat
C) sat

Multiplicative attention:

$$g(h_{i}^{\text{enc}}, h^{\text{dec}}) = (h^{\text{dec}})^T W h_{i}^{\text{enc}}$$
Which value of $W$ in multiplicative attention will provide the same word with highest attention value as dot-product attention?

**Multiplicative attention:**

$$g(h_i^{enc}, h^{dec}) = (h^{dec})^T W h_i^{enc}$$

A) $W = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$  
B) $W = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$  
C) both
## Attention improves translation

<table>
<thead>
<tr>
<th>System</th>
<th>Ppl</th>
<th>BLEU</th>
</tr>
</thead>
<tbody>
<tr>
<td>Winning WMT’14 system – <em>phrase-based + large LM</em> (Buck et al., 2014)</td>
<td></td>
<td>20.7</td>
</tr>
<tr>
<td><strong>Existing NMT systems</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RNNsearch (Jean et al., 2015)</td>
<td>16.5</td>
<td></td>
</tr>
<tr>
<td>RNNsearch + unk replace (Jean et al., 2015)</td>
<td>19.0</td>
<td></td>
</tr>
<tr>
<td>RNNsearch + unk replace + large vocab + <em>ensemble 8 models</em> (Jean et al., 2015)</td>
<td>21.6</td>
<td></td>
</tr>
<tr>
<td><strong>Our NMT systems</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base</td>
<td>10.6</td>
<td>11.3</td>
</tr>
<tr>
<td>Base + reverse</td>
<td>9.9</td>
<td>12.6 (+1.3)</td>
</tr>
<tr>
<td>Base + reverse + dropout</td>
<td>8.1</td>
<td>14.0 (+1.4)</td>
</tr>
<tr>
<td>Base + reverse + dropout + global attention (<em>location</em>)</td>
<td>7.3</td>
<td>16.8 (+2.8)</td>
</tr>
<tr>
<td>Base + reverse + dropout + global attention (<em>location</em>) + feed input</td>
<td>6.4</td>
<td>18.1 (+1.3)</td>
</tr>
<tr>
<td>Base + reverse + dropout + local-p attention (<em>general</em>) + feed input</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Base + reverse + dropout + local-p attention (<em>general</em>) + feed input + unk replace</td>
<td>5.9</td>
<td>19.0 (+0.9)</td>
</tr>
<tr>
<td><em>Ensemble 8 models + unk replace</em></td>
<td></td>
<td>23.0 (+2.1)</td>
</tr>
</tbody>
</table>

(Luong et al., 2015)
Visualizing attention
Going all in on attention

• More recent models (e.g. Transformer, Vaswani et al., 2017) have replaced RNNs entirely with attention mechanisms

• Theoretically limiting (since recurrence can help handle arbitrarily long sequences)

• Huge gains in practical performance
WMT 2014, English-German
Issues with vanilla seq2seq

- A single encoding vector, $h^{enc}$, needs to capture all the information about source sentence
- Longer sequences can lead to vanishing gradients
- Model may “overfit” to training sequences
## Dropout

- Form of regularization for RNNs (and any NN in general)
- **Idea:** "Handicap" NN by removing hidden units *stochastically*
  - set each hidden unit in a layer to 0 with probability $p$ during training ($p = 0.5$ usually works well)
  - scale outputs by $1/(1 - p)$
  - hidden units forced to learn more general patterns and improve redundancy
- **Test time:** Simply compute identity

(Srivastava et al., 2014)
<table>
<thead>
<tr>
<th>ID</th>
<th>System</th>
<th>BLEU 5k</th>
<th>BLEU 10k</th>
<th>BLEU 20k</th>
<th>BLEU 40k</th>
<th>BLEU 80k</th>
<th>BLEU 165k</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Transformer-big</td>
<td>3.3</td>
<td>3.4</td>
<td>4.3</td>
<td>4.7</td>
<td>5.1</td>
<td>5.5</td>
</tr>
<tr>
<td>2</td>
<td>Transformer-base</td>
<td>8.3</td>
<td>11.9</td>
<td>16.8</td>
<td>23.2</td>
<td>28.0</td>
<td>32.1</td>
</tr>
<tr>
<td>3</td>
<td>2 + feed-forward dimension (2048 → 512)</td>
<td>8.8</td>
<td>12.0</td>
<td>16.7</td>
<td>22.3</td>
<td>27.7</td>
<td>31.7</td>
</tr>
<tr>
<td>4</td>
<td>3 + attention heads (8→2)</td>
<td>9.2</td>
<td>12.7</td>
<td>19.0</td>
<td>23.6</td>
<td>28.7</td>
<td>32.3</td>
</tr>
<tr>
<td>5</td>
<td>4 + dropout (0.1→0.3)</td>
<td>10.6</td>
<td>17.0</td>
<td>21.9</td>
<td>26.7</td>
<td><strong>31.0</strong></td>
<td><strong>33.4</strong></td>
</tr>
<tr>
<td>6</td>
<td>5 + layers (6 → 5)</td>
<td>10.9</td>
<td>16.9</td>
<td>21.9</td>
<td>26.0</td>
<td>30.2</td>
<td>33.0</td>
</tr>
<tr>
<td>7</td>
<td>6 + label smoothing (0.1→0.6)</td>
<td>11.3</td>
<td>16.5</td>
<td>22.0</td>
<td>26.9</td>
<td>30.4</td>
<td>33.3</td>
</tr>
<tr>
<td>8</td>
<td>7 + decoder layerDrop (0 → 0.3)</td>
<td>12.9</td>
<td>17.3</td>
<td>22.5</td>
<td>26.9</td>
<td>30.3</td>
<td>33.1</td>
</tr>
<tr>
<td>9</td>
<td>8 + target word dropout (0 → 0.1)</td>
<td>13.7</td>
<td>18.1</td>
<td>23.1</td>
<td>27.0</td>
<td>30.7</td>
<td>33.0</td>
</tr>
<tr>
<td>10</td>
<td>9 + activation dropout (0 → 0.3)</td>
<td><strong>14.3</strong></td>
<td><strong>18.3</strong></td>
<td><strong>23.6</strong></td>
<td><strong>27.4</strong></td>
<td>30.4</td>
<td>32.6</td>
</tr>
</tbody>
</table>

Table 2: Results of Transformer optimized on the 5k dataset for different subsets and full corpus of IWSLT14 German → English. Averages over three runs from three different samples are reported.

(Araabi and Monz, 2020)
Other challenges with NMT

- Out-of-vocabulary words
- Low-resource languages
- Long-term context
- Common sense knowledge (e.g. *hot dog*, *paper jam*)
- Fairness and bias
- Uninterpretable
Massively multilingual MT

- Train a single neural network on 103 languages paired with English (remember Interlingua?)
- Massive improvements on low-resource languages

(Arivazhagan et al., 2019)
Ő szép. Ő okos. Ő olvas. Ő mosogat. Ő épít. Ő varr. Ő tanít. Ő főz. Ő kutat. Ő gyereket nevel. Ő zenél. Ő takarító. Ő politikus. Ő sok pénzt keres. Ő süteményt süt. Ő professzor. Ő asszisztens.

She is beautiful. He is clever. He reads. She washes the dishes. He builds. She sews. He teaches. She cooks. He's researching. She is raising a child. He plays music. She's a cleaner. He is a politician. He makes a lot of money. She is baking a cake. He's a professor. She's an assistant.

Hindi: वो सुंदर है. वो बुद्धिमान है. वो पढ़ाकू है. वो व्यस्त है. वो अमीर है.

English: She is beautiful. He is intelligent. She is busy. He is rich.
Bias and Fairness

- NMT systems suffer from issues of systematic bias (e.g. gender)
- Evident when translating from/to a language with gender-specific (or gender-agnostic) terms
- Models learn (and amplify) stereotypes from data

(Farkas and Nemeth, 2020)
Measuring bias in MT

- WinoMT: Stanovsky et al. (2019) use coreference resolution to construct a dataset of non-stereotypical gender roles
  - e.g. “The doctor asked the nurse to help her in the operation”
- Systems consistently performed worse on non-stereotypical gender translation
<table>
<thead>
<tr>
<th>Source</th>
<th>[Target lang.] Predicted translation</th>
<th>Phenomenon</th>
</tr>
</thead>
<tbody>
<tr>
<td>The janitor does not like <strong>the baker</strong> because <strong>she</strong> always messes up the kitchen.</td>
<td>[ES] Al conserje no le gusta <strong>el panadero</strong> porque <strong>ella</strong> siempre desordena la cocina.</td>
<td>Biased translation, giving “baker” a male inflection, with a mismatched pronoun reference.</td>
</tr>
<tr>
<td>The janitor does not like <strong>the pretty baker</strong> because <strong>she</strong> always messes up the kitchen.</td>
<td>[ES] Al conserje no le gusta <strong>la panadera bonita</strong> porque <strong>ella</strong> siempre desordena la cocina.</td>
<td>Adding a stereotypically female adjective “fixes” the translation.</td>
</tr>
<tr>
<td>The counselor asked <strong>the guard</strong> a few questions and praised <strong>her</strong> for the good work.</td>
<td>[FR] Le conseiller a posé quelques questions à <strong>la garde</strong> et l’a louée pour le bon travail.</td>
<td>French uses “garde” for both male and female guards, allowing for a more direct translation from English.</td>
</tr>
</tbody>
</table>

Table 5: Examples of Google Translate’s output for different sentences in the WinoMT corpus. Words in **blue**, **red**, and **orange** indicate male, female and neutral entities, respectively.
Mitigating bias

- Stafanovics et al. (2020) use word-level annotations of subject’s gender to train NMT systems

- TGA (target gender annotations) help reduce gender bias ($\nabla G =$ diff. in F1 between sentences with male and female antecedents, $\nabla S =$ diff. in accuracy between sentences w/ or w/o stereotypes)

### WMT Data Systems

<table>
<thead>
<tr>
<th></th>
<th>Acc.</th>
<th>$\Delta G$</th>
<th>$\Delta S$</th>
<th>M:F</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>EN-DE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>66.7</td>
<td>10.2</td>
<td>14.4</td>
<td>2.6</td>
</tr>
<tr>
<td>TGA Oracle</td>
<td>89.0</td>
<td>-4.7</td>
<td>1.7</td>
<td>1</td>
</tr>
<tr>
<td>TGA HuggingFace</td>
<td>77.6</td>
<td>-0.1</td>
<td>11.9</td>
<td>1.6</td>
</tr>
<tr>
<td>TGA AllenNLP</td>
<td>81.5</td>
<td>-2.0</td>
<td>11.1</td>
<td>1.4</td>
</tr>
<tr>
<td><strong>EN-FR</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Baseline</td>
<td>48.6</td>
<td>29.8</td>
<td>11.8</td>
<td>5.5</td>
</tr>
<tr>
<td>TGA Oracle</td>
<td>81.5</td>
<td>1.4</td>
<td>2.8</td>
<td>1.2</td>
</tr>
<tr>
<td>TGA HuggingFace</td>
<td>67.8</td>
<td>4.9</td>
<td>12.4</td>
<td>2</td>
</tr>
<tr>
<td>TGA AllenNLP</td>
<td>74.4</td>
<td>1.6</td>
<td>10.1</td>
<td>1.6</td>
</tr>
</tbody>
</table>
Anonymous feedback form:
https://forms.gle/7BxYDUTebogndJQE8