Edinburgh’s WMT Results Over the Years

![Bar chart showing BLEU scores over the years for phrase-based SMT, syntax-based SMT, and neural MT.]

- **2013:**
  - Phrase-based SMT: 20.3
  - Syntax-based SMT: 19.4

- **2014:**
  - Phrase-based SMT: 20.9
  - Syntax-based SMT: 20.2

- **2015:**
  - Phrase-based SMT: 20.8
  - Syntax-based SMT: 22.0
  - Neural MT: 18.9

- **2016:**
  - Phrase-based SMT: 21.5
  - Syntax-based SMT: 22.1
  - Neural MT: 24.7

(NMT 2015 from U. Montréal: [https://sites.google.com/site/acl16nmt/](https://sites.google.com/site/acl16nmt/))

Rico Sennrich
\[ f = (\text{La, croissance, économique, s'est, ralentie, ces, dernières, années, .}) \]

\[ e = (\text{Economic, growth, has, slowed, down, in, recent, years, .}) \]

\[ \sum a_j = 1 \]

Kyunghyun Cho
1. Attentional encoder-decoder

2. Where are we now? Evaluation, challenges, future directions...
   - Evaluation results
   - Comparing neural and phrase-based machine translation
   - Recent research in neural machine translation
Translation modelling

decomposition of translation problem (for NMT)

- a source sentence $S$ of length $m$ is a sequence $x_1, \ldots, x_m$
- a target sentence $T$ of length $n$ is a sequence $y_1, \ldots, y_n$

\[
T^* = \arg \max_{t} p(T|S)
\]

\[
p(T|S) = p(y_1, \ldots, y_n|x_1, \ldots, x_m)
\]

\[
= \prod_{i=1}^{n} p(y_i|y_0, \ldots, y_{i-1}, x_1, \ldots, x_m)
\]
Translation modelling

difference from language model

- target-side language model:

\[ p(T) = \prod_{i=1}^{n} p(y_i | y_0, \ldots, y_{i-1}) \]

- translation model:

\[ p(T | S) = \prod_{i=1}^{n} p(y_i | y_0, \ldots, y_{i-1}, x_1, \ldots, x_m) \]

- we could just treat sentence pair as one long sequence, but:
  - we do not care about \( p(S) \) (\( S \) is given)
  - we do not want to use same parameters for \( S \) and \( T \)
  - we may want different vocabulary, network architecture for source text
Translation modelling

difference from language model

- target-side language model:

$$p(T) = \prod_{i=1}^{n} p(y_i | y_0, \ldots, y_{i-1})$$

- translation model:

$$p(T | S) = \prod_{i=1}^{n} p(y_i | y_0, \ldots, y_{i-1}, x_1, \ldots, x_m)$$

- we could just treat sentence pair as one long sequence, but:
  - we do not care about $p(S)$ ($S$ is given)
  - we do not want to use same parameters for $S$ and $T$
  - we may want different vocabulary, network architecture for source text
Encoder-decoder [Sutskever et al., 2014, Cho et al., 2014]

- two RNNs (LSTM or GRU):
  - **encoder** reads input and produces hidden state representations
  - **decoder** produces output, based on last encoder hidden state
- joint learning (backpropagation through full network)
Summary vector

- last encoder hidden-state “summarizes” source sentence
- with multilingual training, we can potentially learn language-independent meaning representation

[Sutskever et al., 2014]
Summary vector as information bottleneck

- Can fixed-size vector represent meaning of arbitrarily long sentence?
- Empirically, quality decreases for long sentences
- Reversing source sentence brings some improvement
  [Sutskever et al., 2014]
encoder

- **goal**: avoid bottleneck of summary vector
- use bidirectional RNN, and concatenate forward and backward states
  \[ \rightarrow \text{annotation vector } h_i \]
- represent source sentence as vector of \( n \) annotations
  \[ \rightarrow \text{variable-length representation} \]
Attentional encoder-decoder [Bahdanau et al., 2015]

**Attention**

- problem: how to incorporate variable-length context into hidden state?
- *attention model* computes *context vector* as weighted average of annotations
- weights are computed by feedforward neural network with softmax

\[
f = (\text{La, croissance, économiue, s'est, ralentie, ces, dernières, années, .})
\]

\[
f = (\text{Economic, growth, has, slowed, down, in, recent, years, .})
\]

Kyunghyun Cho
Attentional encoder-decoder: math

- simplifications of model by [Bahdanau et al., 2015] (for illustration)
  - plain RNN instead of GRU
  - simpler output layer
  - we do not show bias terms

- notation
  - \( W, U, E, C, V \) are weight matrices (of different dimensionality)
    - \( E \) one-hot to embedding (e.g. 50000 \( \cdot \) 512)
    - \( W \) embedding to hidden (e.g. 512 \( \cdot \) 1024)
    - \( U \) hidden to hidden (e.g. 1024 \( \cdot \) 1024)
    - \( C \) context (2x hidden) to hidden (e.g. 2048 \( \cdot \) 1024)
    - \( V_o \) hidden to one-hot (e.g. 1024 \( \cdot \) 50000)
  - separate weight matrices for encoder and decoder (e.g. \( E_x \) and \( E_y \))
  - input \( X \) of length \( T_x \); output \( Y \) of length \( T_y \)
Attentional encoder-decoder: math

**encoder**

\[
\begin{align*}
\overrightarrow{h}_j &= \begin{cases} 
0, & \text{if } j = 0 \\
\tanh(\overrightarrow{W}_x E_x x_j + \overrightarrow{U}_x \overrightarrow{h}_{j-1}) & \text{if } j > 0 
\end{cases} \\
\overleftarrow{h}_j &= \begin{cases} 
0, & \text{if } j = T_x + 1 \\
\tanh(\overleftarrow{W}_x E_x x_j + \overleftarrow{U}_x \overleftarrow{h}_{j+1}) & \text{if } j \leq T_x 
\end{cases} \\
h_j &= (\overrightarrow{h}_j, \overleftarrow{h}_j)
\end{align*}
\]
Attentional encoder-decoder: math

decoder

\[
\begin{align*}
s_i &= \begin{cases} 
\tanh(W_s h_i), & \text{if } i = 0 \\
\tanh(W_y E_y y_i + U_y s_{i-1} + C c_i) & \text{if } i > 0
\end{cases} \\
t_i &= \tanh(U_o s_{i-1} + W_o E_y y_{i-1} + C_o c_i) \\
y_i &= \text{softmax}(V_o t_i)
\end{align*}
\]

attention model

\[
\begin{align*}
e_{ij} &= v_a^\top \tanh(W_a s_{i-1} + U_a h_j) \\
\alpha_{ij} &= \text{softmax}(e_{ij}) \\
c_i &= \sum_{j=1}^{T_x} \alpha_{ij} h_j
\end{align*}
\]
Attention model

- side effect: we obtain alignment between source and target sentence information can also flow along recurrent connections, so there is no guarantee that attention corresponds to alignment

- applications:
  - visualisation
  - replace unknown words with back-off dictionary [Jean et al., 2015]
  - ...

Economic growth has slowed down in recent years.

Das Wirtschaftswachstum hat sich in den letzten Jahren verlangsamt.

La croissance économique s’est ralentie ces dernières années.
Attention model

attention model also works with images:

\[ f = (a, \text{ man, is, jumping, into, a, lake, .}) \]

[Cho et al., 2015]
Fig. 5. Examples of the attention-based model attending to the correct object (*white* indicates the attended regions, *underlines* indicated the corresponding word) [22]
Applications of encoder-decoder neural network

score a translation

\[ p(\text{La, croissance, économique, s’est, ralentie, ces, dernières, années, . | Economic, growth, has, slowed, down, in, recent, year, .}) = \] ?

generate the most probable translation of a source sentence → decoding

\[ y^* = \arg\max_{y} p( y | \text{Economic, growth, has, slowed, down, in, recent, year, .} ) \]
Decoding

### exact search

- generate every possible sentence $T$ in target language
- compute score $p(T|S)$ for each
- pick best one

- intractable: $|V|^N$ translations for vocabulary $V$ and output length $N$
  $\rightarrow$ we need approximative search strategy
Decoding

approximative search/1

- at each time step, compute probability distribution $P(y_i | X, y_{<i})$
- select $y_i$ according to some heuristic:
  - sampling: sample from $P(y_i | X, y_{<i})$
  - greedy search: pick $\arg\max_y p(y_i | X, y_{<i})$
- continue until we generate $<eos>$

- efficient, but suboptimal
approximative search/2: **beam search**

- maintain list of $K$ hypotheses (beam)
- at each time step, expand each hypothesis $k$: $p(y^k_i | X, y^k_{<i})$
- at each time step, we produce $|V| \cdot K$ translation hypotheses
  → prune to $K$ hypotheses with highest total probability:

$$\prod_i p(y^k_i | X, y^k_{<i})$$

- relatively efficient
- currently default search strategy in neural machine translation
- small beam ($K \approx 10$) offers good speed-quality trade-off
Ensembles

- at each timestep, combine the probability distribution of $M$ different ensemble components.
- combine operator: typically average (log-)probability

$$\log P(y_i|X, y_{<i}) = \frac{\sum_{m=1}^{M} \log P_m(y_i|X, y_{<i})}{M}$$

- requirements:
  - same output vocabulary
  - same factorization of $Y$
- internal network architecture may be different
- source representations may be different
  (extreme example: ensemble-like model with different source languages [Junczys-Dowmunt and Grundkiewicz, 2016])
recent ensemble strategies in NMT

- ensemble of 8 independent training runs with different hyperparameters/architectures [Luong et al., 2015a]
- ensemble of 8 independent training runs with different random initializations [Chung et al., 2016]
- ensemble of 4 checkpoints of same training run [Sennrich et al., 2016a]
  → probably less effective, but only requires one training run
1. Attentional encoder-decoder

2. Where are we now? Evaluation, challenges, future directions...
   - Evaluation results
   - Comparing neural and phrase-based machine translation
   - Recent research in neural machine translation
attentional encoder-decoder networks have become state of the art on various MT tasks...

...but this usually requires more advanced techniques to handle OOVs, use monolingual data, etc.

your mileage may vary depending on
- language pair and text type
- amount of training data
- type of training resources (monolingual?)
- hyperparameters

very general model: can be applied to other sequence-to-sequence tasks
Attentional encoder-decoders (NMT) are SOTA

Table: WMT16 results for EN→DE

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>34.2</td>
<td>1</td>
</tr>
<tr>
<td>metamind</td>
<td>32.3</td>
<td>2</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>30.6</td>
<td>3</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>30.8</td>
<td>4</td>
</tr>
<tr>
<td>online-B</td>
<td>29.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT/LIMSI</td>
<td>29.1</td>
<td>5-10</td>
</tr>
<tr>
<td>cambridge</td>
<td>30.6</td>
<td>5-10</td>
</tr>
<tr>
<td>online-A</td>
<td>29.9</td>
<td>5-10</td>
</tr>
<tr>
<td>prompt-rule</td>
<td>23.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT</td>
<td>29.0</td>
<td>6-10</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>26.6</td>
<td>11-12</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>28.3</td>
<td>11-12</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>28.4</td>
<td>13-14</td>
</tr>
<tr>
<td>online-F</td>
<td>19.3</td>
<td>13-15</td>
</tr>
<tr>
<td>online-G</td>
<td>23.8</td>
<td>14-15</td>
</tr>
</tbody>
</table>

Table: WMT16 results for DE→EN

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>38.6</td>
<td>1</td>
</tr>
<tr>
<td>online-B</td>
<td>35.0</td>
<td>2-5</td>
</tr>
<tr>
<td>online-A</td>
<td>32.8</td>
<td>2-5</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>34.4</td>
<td>2-5</td>
</tr>
<tr>
<td>KIT</td>
<td>33.9</td>
<td>2-6</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>35.1</td>
<td>5-7</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>34.5</td>
<td>6-7</td>
</tr>
<tr>
<td>online-G</td>
<td>30.1</td>
<td>8</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>31.0</td>
<td>9</td>
</tr>
<tr>
<td>online-F</td>
<td>20.2</td>
<td>10</td>
</tr>
</tbody>
</table>
Attentional encoder-decoders (NMT) are SOTA

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>34.2</td>
<td>1</td>
</tr>
<tr>
<td>metamind</td>
<td>32.3</td>
<td>2</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>30.6</td>
<td>3</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>30.8</td>
<td>4</td>
</tr>
<tr>
<td>online-B</td>
<td>29.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT/LIMSI</td>
<td>29.1</td>
<td>5-10</td>
</tr>
<tr>
<td>cambridge</td>
<td>30.6</td>
<td>5-10</td>
</tr>
<tr>
<td>online-A</td>
<td>29.9</td>
<td>5-10</td>
</tr>
<tr>
<td>prompt-rule</td>
<td>23.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT</td>
<td>29.0</td>
<td>6-10</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>26.6</td>
<td>11-12</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>28.3</td>
<td>11-12</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>28.4</td>
<td>13-14</td>
</tr>
<tr>
<td>online-F</td>
<td>19.3</td>
<td>13-15</td>
</tr>
<tr>
<td>online-G</td>
<td>23.8</td>
<td>14-15</td>
</tr>
</tbody>
</table>

Table: WMT16 results for EN→DE

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>38.6</td>
<td>1</td>
</tr>
<tr>
<td>online-B</td>
<td>35.0</td>
<td>2-5</td>
</tr>
<tr>
<td>online-A</td>
<td>32.8</td>
<td>2-5</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>34.4</td>
<td>2-5</td>
</tr>
<tr>
<td>KIT</td>
<td>33.9</td>
<td>2-6</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>35.1</td>
<td>5-7</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>34.5</td>
<td>6-7</td>
</tr>
<tr>
<td>online-G</td>
<td>30.1</td>
<td>8</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>31.0</td>
<td>9</td>
</tr>
<tr>
<td>online-F</td>
<td>20.2</td>
<td>10</td>
</tr>
</tbody>
</table>

Table: WMT16 results for DE→EN

- pure NMT
Attentional encoder-decoders (NMT) are SOTA

### Table: WMT16 results for EN → DE

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>34.2</td>
<td>1</td>
</tr>
<tr>
<td>metamind</td>
<td>32.3</td>
<td>2</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>30.6</td>
<td>3</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>30.8</td>
<td>4</td>
</tr>
<tr>
<td>online-B</td>
<td>29.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT/LIMSI</td>
<td>29.1</td>
<td>5-10</td>
</tr>
<tr>
<td>cambridge</td>
<td>30.6</td>
<td>5-10</td>
</tr>
<tr>
<td>online-A</td>
<td>29.9</td>
<td>5-10</td>
</tr>
<tr>
<td>promt-rule</td>
<td>23.4</td>
<td>5-10</td>
</tr>
<tr>
<td>KIT</td>
<td>29.0</td>
<td>6-10</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>26.6</td>
<td>11-12</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>28.3</td>
<td>11-12</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>28.4</td>
<td>13-14</td>
</tr>
<tr>
<td>online-F</td>
<td>19.3</td>
<td>13-15</td>
</tr>
<tr>
<td>online-G</td>
<td>23.8</td>
<td>14-15</td>
</tr>
</tbody>
</table>

### Table: WMT16 results for DE → EN

<table>
<thead>
<tr>
<th>system</th>
<th>BLEU</th>
<th>official rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>38.6</td>
<td>1</td>
</tr>
<tr>
<td>online-B</td>
<td>35.0</td>
<td>2-5</td>
</tr>
<tr>
<td>online-A</td>
<td>32.8</td>
<td>2-5</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>34.4</td>
<td>2-5</td>
</tr>
<tr>
<td>KIT</td>
<td>33.9</td>
<td>2-6</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>35.1</td>
<td>5-7</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>34.5</td>
<td>6-7</td>
</tr>
<tr>
<td>online-G</td>
<td>30.1</td>
<td>8</td>
</tr>
<tr>
<td>jhu-syntax</td>
<td>31.0</td>
<td>9</td>
</tr>
<tr>
<td>online-F</td>
<td>20.2</td>
<td>10</td>
</tr>
</tbody>
</table>

- pure NMT
- NMT component
Attentional encoder-decoders (NMT) are SOTA

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>25.8</td>
<td>1</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>23.6</td>
<td>2</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>23.6</td>
<td>3</td>
</tr>
<tr>
<td>cu-chimera</td>
<td>21.0</td>
<td>4-5</td>
</tr>
<tr>
<td>cu-tamchyna</td>
<td>20.8</td>
<td>4-5</td>
</tr>
<tr>
<td>uedin-cu-syntax</td>
<td>20.9</td>
<td>6-7</td>
</tr>
<tr>
<td>online-B</td>
<td>22.7</td>
<td>6-7</td>
</tr>
<tr>
<td>online-A</td>
<td>19.5</td>
<td>15</td>
</tr>
<tr>
<td>cu-TectoMT</td>
<td>14.7</td>
<td>16</td>
</tr>
<tr>
<td>cu-mergedtrees</td>
<td>8.2</td>
<td>18</td>
</tr>
</tbody>
</table>

Table: WMT16 results for EN→CS

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>online-B</td>
<td>39.2</td>
<td>1-2</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>33.9</td>
<td>1-2</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>35.2</td>
<td>3</td>
</tr>
<tr>
<td>uedin-syntax</td>
<td>33.6</td>
<td>4-5</td>
</tr>
<tr>
<td>online-A</td>
<td>30.8</td>
<td>4-6</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>32.2</td>
<td>5-7</td>
</tr>
<tr>
<td>LIMSI</td>
<td>31.0</td>
<td>6-7</td>
</tr>
</tbody>
</table>

Table: WMT16 results for CS→EN

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>31.4</td>
<td>1</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>30.4</td>
<td>2</td>
</tr>
<tr>
<td>online-B</td>
<td>28.6</td>
<td>3</td>
</tr>
<tr>
<td>PJATK</td>
<td>28.3</td>
<td>8-10</td>
</tr>
<tr>
<td>online-A</td>
<td>25.7</td>
<td>11</td>
</tr>
<tr>
<td>cu-mergedtrees</td>
<td>13.3</td>
<td>12</td>
</tr>
</tbody>
</table>

Table: WMT16 results for RO→EN

<table>
<thead>
<tr>
<th>Method</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>28.1</td>
<td>1-2</td>
</tr>
<tr>
<td>QT21-HimL-SysComb</td>
<td>28.9</td>
<td>1-2</td>
</tr>
<tr>
<td>KIT</td>
<td>25.8</td>
<td>3-7</td>
</tr>
<tr>
<td>uedin-pbmt</td>
<td>26.8</td>
<td>3-7</td>
</tr>
<tr>
<td>online-B</td>
<td>25.4</td>
<td>3-7</td>
</tr>
<tr>
<td>uedin-lmu-hiero</td>
<td>25.9</td>
<td>3-7</td>
</tr>
<tr>
<td>RWTH-SYSCOMB</td>
<td>27.1</td>
<td>3-7</td>
</tr>
<tr>
<td>LIMSI</td>
<td>23.9</td>
<td>8-10</td>
</tr>
<tr>
<td>lmu-cuni</td>
<td>24.3</td>
<td>8-10</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>23.5</td>
<td>8-11</td>
</tr>
<tr>
<td>usfd-rescoring</td>
<td>23.1</td>
<td>10-12</td>
</tr>
<tr>
<td>online-A</td>
<td>19.2</td>
<td>11-12</td>
</tr>
</tbody>
</table>

Table: WMT16 results for EN→RO

Rico Sennrich

Neural Machine Translation

25/65
Attentional encoder-decoders (NMT) are SOTA

<table>
<thead>
<tr>
<th>System</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>amu-uedin</td>
<td>25.3</td>
<td>2-4</td>
</tr>
<tr>
<td>online-B</td>
<td>23.8</td>
<td>2-5</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>26.0</td>
<td>2-5</td>
</tr>
<tr>
<td>online-G</td>
<td>26.2</td>
<td>3-5</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>23.1</td>
<td>6</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>24.0</td>
<td>7-8</td>
</tr>
<tr>
<td>LIMSI</td>
<td>23.6</td>
<td>7-10</td>
</tr>
<tr>
<td>online-A</td>
<td>20.2</td>
<td>8-10</td>
</tr>
<tr>
<td>AFRL-MITLL-phr</td>
<td>23.5</td>
<td>9-10</td>
</tr>
<tr>
<td>AFRL-MITLL-verb</td>
<td>20.9</td>
<td>11</td>
</tr>
<tr>
<td>online-F</td>
<td>8.6</td>
<td>12</td>
</tr>
</tbody>
</table>

Table: WMT16 results for EN→RU

<table>
<thead>
<tr>
<th>System</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>amu-uedin</td>
<td>29.1</td>
<td>1-2</td>
</tr>
<tr>
<td>online-G</td>
<td>28.7</td>
<td>1-3</td>
</tr>
<tr>
<td>NRC</td>
<td>29.1</td>
<td>2-4</td>
</tr>
<tr>
<td>online-B</td>
<td>28.1</td>
<td>3-5</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>28.0</td>
<td>4-5</td>
</tr>
<tr>
<td>online-A</td>
<td>25.7</td>
<td>6-7</td>
</tr>
<tr>
<td>AFRL-MITLL-phr</td>
<td>27.6</td>
<td>6-7</td>
</tr>
<tr>
<td>AFRL-MITLL-contrast</td>
<td>27.0</td>
<td>8-9</td>
</tr>
<tr>
<td>PROMT-rule</td>
<td>20.4</td>
<td>8-9</td>
</tr>
<tr>
<td>online-F</td>
<td>13.5</td>
<td>10</td>
</tr>
</tbody>
</table>

Table: WMT16 results for RU→EN

<table>
<thead>
<tr>
<th>System</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>abumatra-nmt</td>
<td>17.2</td>
<td>1-4</td>
</tr>
<tr>
<td>online-B</td>
<td>14.4</td>
<td>1-4</td>
</tr>
<tr>
<td>abumatran-combo</td>
<td>17.4</td>
<td>3-5</td>
</tr>
<tr>
<td>UH-opus</td>
<td>16.3</td>
<td>4-5</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>15.1</td>
<td>6-8</td>
</tr>
<tr>
<td>abumatran-pbsmt</td>
<td>14.6</td>
<td>6-8</td>
</tr>
<tr>
<td>online-A</td>
<td>13.0</td>
<td>6-8</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>13.8</td>
<td>9-10</td>
</tr>
<tr>
<td>UH-factored</td>
<td>12.8</td>
<td>9-12</td>
</tr>
<tr>
<td>aalto</td>
<td>11.6</td>
<td>10-13</td>
</tr>
<tr>
<td>jhu-hltcoe</td>
<td>11.9</td>
<td>10-13</td>
</tr>
<tr>
<td>UUT</td>
<td>11.6</td>
<td>11-13</td>
</tr>
</tbody>
</table>

Table: WMT16 results for FI→EN

<table>
<thead>
<tr>
<th>System</th>
<th>Score</th>
<th>Rank</th>
</tr>
</thead>
<tbody>
<tr>
<td>online-G</td>
<td>15.4</td>
<td>1-3</td>
</tr>
<tr>
<td>abumatra-nmt</td>
<td>17.2</td>
<td>1-4</td>
</tr>
<tr>
<td>online-B</td>
<td>14.4</td>
<td>1-4</td>
</tr>
<tr>
<td>abumatran-combo</td>
<td>17.4</td>
<td>3-5</td>
</tr>
<tr>
<td>UH-opus</td>
<td>16.3</td>
<td>4-5</td>
</tr>
<tr>
<td>NYU-UMontreal</td>
<td>15.1</td>
<td>6-8</td>
</tr>
<tr>
<td>abumatran-pbsmt</td>
<td>14.6</td>
<td>6-8</td>
</tr>
<tr>
<td>online-A</td>
<td>13.0</td>
<td>6-8</td>
</tr>
<tr>
<td>jhu-pbmt</td>
<td>13.8</td>
<td>9-10</td>
</tr>
<tr>
<td>UH-factored</td>
<td>12.8</td>
<td>9-12</td>
</tr>
<tr>
<td>aalto</td>
<td>11.6</td>
<td>10-13</td>
</tr>
<tr>
<td>jhu-hltcoe</td>
<td>11.9</td>
<td>10-13</td>
</tr>
<tr>
<td>UUT</td>
<td>11.6</td>
<td>11-13</td>
</tr>
</tbody>
</table>

Table: WMT16 results for EN→FI
1. Attentional encoder-decoder

2. Where are we now? Evaluation, challenges, future directions...
   - Evaluation results
   - Comparing neural and phrase-based machine translation
   - Recent research in neural machine translation
ambiguity

words are often polysemous, with different translations for different meanings

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Dort wurde er von dem Schläger und einer weiteren männlichen Person erneut angegriffen.</td>
</tr>
<tr>
<td>reference</td>
<td>There he was attacked again by his original attacker and another male.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>There, he was at the club and another male person attacked again.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>There he was attacked again by the racket and another male person.</td>
</tr>
</tbody>
</table>

Schläger
Interlude: why is (machine) translation hard?

ambiguity

words are often polysemous, with different translations for different meanings

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Dort wurde er von dem Schläger und einer weiteren männlichen Person erneut angegriffen.</td>
</tr>
<tr>
<td>reference</td>
<td>There he was attacked again by his original attacker and another male.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>There, he was at the club and another male person attacked again.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>There he was attacked again by the racket and another male person.</td>
</tr>
</tbody>
</table>

Schläger

racket
Interlude: why is (machine) translation hard?

ambiguity

words are often polysemous, with different translations for different meanings

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Dort wurde er von dem Schläger und einer weiteren männlichen Person erneut angegriffen.</td>
</tr>
<tr>
<td>reference</td>
<td>There he was attacked again by his original attacker and another male.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>There, he was at the club and another male person attacked again.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>There he was attacked again by the racket and another male person.</td>
</tr>
</tbody>
</table>
Interlude: why is (machine) translation hard?

**ambiguity**

Words are often polysemous, with different translations for different meanings.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Dort wurde er von dem Schläger und einer weiteren männlichen Person erneut angegriffen.</td>
</tr>
<tr>
<td>reference</td>
<td>There he was attacked again by his original attacker and another male.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>There, he was at the club and another male person attacked again.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>There he was attacked again by the racket and another male person.</td>
</tr>
</tbody>
</table>

---

Schläger

- Racket
- Attacker
- Club
Interlude: why is (machine) translation hard?

---

**word order**

there are systematic word order differences between languages. We need to generate words in the correct order.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source reference</td>
<td>Unsere digitalen Leben <em>haben</em> die Notwendigkeit, stark, lebenslustig und erfolgreich zu erscheinen, <em>verdoppelt</em> [...]. Our digital lives <em>have doubled</em> the need to appear strong, fun-loving and successful [...].</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>Our digital lives <em>are</em> lively, strong, and to be successful, <em>doubled</em> [...].</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>Our digital lives <em>have doubled</em> the need to appear strong, lifelike and successful [...].</td>
</tr>
</tbody>
</table>
Interlude: why is (machine) translation hard?

grammatical marking system

grammatical distinctions can be marked in different ways, for instance through word order (English), or inflection (German). The translator needs to produce the appropriate marking.

English ... because the dog chased the man.
German ... weil der Hund den Mann jagte.
### multiword expressions

The meaning of non-compositional expressions is lost in a word-to-word translation.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>He <em>bends over backwards</em> for the team, ignoring any pain.</td>
</tr>
<tr>
<td></td>
<td>Er <em>zerreißt sich</em> für die Mannschaft, geht über Schmerzen drüber. (lit: he tears himself apart for the team)</td>
</tr>
<tr>
<td>reference</td>
<td></td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>Er <em>macht alles</em> für das Team, den Schmerz zu ignorieren. (lit: he does everything for the team)</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>Er <em>beugt sich rückwärts</em> für die Mannschaft, ignoriert jeden Schmerz. (lit: he bends backwards for the team)</td>
</tr>
</tbody>
</table>
### Interlude: why is (machine) translation hard?

#### subcategorization

Words only allow for specific categories of syntactic arguments, that often differ between languages.

<table>
<thead>
<tr>
<th>English</th>
<th>he remembers his medical appointment.</th>
</tr>
</thead>
<tbody>
<tr>
<td>German</td>
<td>er erinnert sich an seinen Arzttermin.</td>
</tr>
<tr>
<td>English</td>
<td>*he remembers himself to his medical appointment.</td>
</tr>
<tr>
<td>German</td>
<td>*er erinnert seinen Arzttermin.</td>
</tr>
</tbody>
</table>

#### agreement

Inflected forms may need to agree over long distances to satisfy grammaticality.

<table>
<thead>
<tr>
<th>English</th>
<th>they can not be found</th>
</tr>
</thead>
<tbody>
<tr>
<td>French</td>
<td>elles ne peuvent pas être trouvées</td>
</tr>
</tbody>
</table>
Interlude: why is (machine) translation hard?

### morphological complexity

Translator may need to analyze/generate morphologically complex words that were not seen before.

<table>
<thead>
<tr>
<th>German</th>
<th>Abwasserbehandlungsanlage</th>
</tr>
</thead>
<tbody>
<tr>
<td>English</td>
<td>waste water treatment plant</td>
</tr>
<tr>
<td>French</td>
<td>station d’épuration des eaux résiduaires</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>System</th>
<th>Sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source</td>
<td>Titelverteidiger ist Drittligaabsteiger SpVgg Unterhaching.</td>
</tr>
<tr>
<td>Reference</td>
<td>The defending champions are SpVgg Unterhaching, who have been relegated to the third league.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>Title defender Drittligaabsteiger Week 2.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>Defending champion is third-round pick SpVgg Underhaching.</td>
</tr>
</tbody>
</table>
Interlude: why is (machine) translation hard?

### open vocabulary

languages have an open vocabulary, and we need to learn translations for words that we have only seen rarely (or never)

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Titelverteidiger ist Drittligaabsteiger SpVgg Unterhaching.</td>
</tr>
<tr>
<td>reference</td>
<td>The defending champions are SpVgg Unterhaching, who have been relegated to the third league.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>Title defender Drittligaabsteiger Week 2.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>Defending champion is third-round pick SpVgg Unterhaching.</td>
</tr>
</tbody>
</table>
Interlude: why is (machine) translation hard?

**discontinuous structures**

A word (sequence) can map to a discontinuous structure in another language.

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source</td>
<td>Ein Jahr später <strong>machten</strong> die Fed-Repräsentanten diese Kürzungen rückgängig.</td>
</tr>
<tr>
<td>reference</td>
<td>A year later, Fed officials <strong>reversed</strong> those cuts.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>A year later, the Fed representatives <strong>made</strong> these cuts.</td>
</tr>
<tr>
<td>uedin-nmt</td>
<td>A year later, FedEx officials <strong>reversed</strong> those cuts.</td>
</tr>
</tbody>
</table>

**English**  I do **not** know

**French**   Je **ne** sais **pas**
### discourse

the translation of referential expressions depends on discourse context, which sentence-level translators have no access to.

<table>
<thead>
<tr>
<th>English</th>
<th>French (formal)</th>
<th>French (informal)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Please respect it.</td>
<td>Respectez-la s’il vous plaît.</td>
<td>Respectez-le s’il vous plaît.</td>
</tr>
</tbody>
</table>
assorted other difficulties

- underspecification
- ellipsis
- lexical gaps
- language change
- language variation (dialects, genres, domains)
- ill-formed input
human analysis of NMT (reranking) [Neubig et al., 2015]

- NMT is more grammatical
  - word order
  - insertion/deletion of function words
  - morphological agreement
- minor degradation in lexical choice?
Comparison between phrase-based and neural MT

Analysis of IWSLT 2015 results [Bentivogli et al., 2016]

- Human-targeted translation error rate (HTER) based on automatic translation and human post-edit
- 4 error types: substitution, insertion, deletion, shift

<table>
<thead>
<tr>
<th>System</th>
<th>HTER (no <em>shift</em>)</th>
<th>HTER (<em>shift only</em>)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>word</td>
<td>lemma</td>
</tr>
<tr>
<td>PBSMT [Ha et al., 2015]</td>
<td>28.3</td>
<td>23.2</td>
</tr>
<tr>
<td>NMT [Luong and Manning, 2015]</td>
<td>21.7</td>
<td>18.7</td>
</tr>
</tbody>
</table>

- Word-level is closer to lemma-level performance: better at inflection/agreement
- Improvement on lemma-level: better lexical choice
- Fewer shift errors: better word order
Comparison between phrase-based and neural MT

Analysis of IWSLT 2015 results [Bentivogli et al., 2016]

- Human-targeted translation error rate (HTER) based on automatic translation and human post-edit
- 4 error types: substitution, insertion, deletion, shift

<table>
<thead>
<tr>
<th>System</th>
<th>HTER (no shift)</th>
<th>HTER (shift only)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>word</td>
<td>lemma</td>
</tr>
<tr>
<td>PBSMT [Ha et al., 2015]</td>
<td>28.3</td>
<td>23.2</td>
</tr>
<tr>
<td>NMT [Luong and Manning, 2015]</td>
<td>21.7</td>
<td>18.7</td>
</tr>
</tbody>
</table>

- Word-level is closer to lemma-level performance: better at inflection/agreement
- Improvement on lemma-level: better lexical choice
- Fewer shift errors: better word order
Comparison between phrase-based and neural MT

**Analysis of IWSLT 2015 results [Bentivogli et al., 2016]**

- Human-targeted translation error rate (HTER) based on automatic translation and human post-edit
- 4 error types: substitution, insertion, deletion, shift

<table>
<thead>
<tr>
<th>System</th>
<th>HTER (no shift)</th>
<th>HTER (shift only)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>word lemma %Δ</td>
<td></td>
</tr>
<tr>
<td>PBSMT [Ha et al., 2015]</td>
<td>28.3 23.2 -18.0</td>
<td>3.5</td>
</tr>
<tr>
<td>NMT [Luong and Manning, 2015]</td>
<td>21.7 18.7 -13.7</td>
<td>1.5</td>
</tr>
</tbody>
</table>

- Word-level is closer to lemma-level performance: better at inflection/agreement
- Improvement on lemma-level: better lexical choice
- Fewer shift errors: better word order
Comparison between phrase-based and neural MT

Analysis of IWSLT 2015 results [Bentivogli et al., 2016]

- Human-targeted translation error rate (HTER) based on automatic translation and human post-edit
- 4 error types: substitution, insertion, deletion, shift

<table>
<thead>
<tr>
<th>System</th>
<th>HTER (no shift)</th>
<th>HTER (shift only)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>word</td>
<td>lemma</td>
</tr>
<tr>
<td>PBSMT [Ha et al., 2015]</td>
<td>28.3</td>
<td>23.2</td>
</tr>
<tr>
<td>NMT [Luong and Manning, 2015]</td>
<td>21.7</td>
<td>18.7</td>
</tr>
</tbody>
</table>

- Word-level is closer to lemma-level performance: better at inflection/agreement
- Improvement on lemma-level: better lexical choice
- Fewer shift errors: better word order
Comparison between phrase-based and neural MT

WMT16 direct assessment [Bojar et al., 2016]

- uedin-nmt is most fluent for all 4 evaluated translation directions
- In adequacy, ranked:
  - 1/6 (CS-EN)
  - 1/10 (DE-EN)
  - 2/7 (RO-EN)
  - 6/10 (RU-EN)
- Relative to other systems, stronger contrast in fluency than adequacy
Why is neural MT output more grammatical?

### neural MT
- end-to-end trained model
- generalization via continuous space representation
- output conditioned on full source text and target history

### phrase-based SMT
- log-linear combination of many “weak” features
- data sparseness triggers back-off to smaller units
- strong independence assumptions
Neural Machine Translation

1. Attentional encoder-decoder

2. Where are we now? Evaluation, challenges, future directions...
   - Evaluation results
   - Comparing neural and phrase-based machine translation
   - Recent research in neural machine translation
Efficiency

speed bottlenecks

- matrix multiplication
  → use of highly parallel hardware (GPUs)
- softmax (scales with vocabulary size). Solutions:
  - LMs: hierarchical softmax; noise-contrastive estimation; self-normalization
  - NMT: approximate softmax through subset of vocabulary [Jean et al., 2015]

NMT training vs. decoding (on fast GPU)

- training: slow (1-3 weeks)
- decoding: fast (100 000–500 000 sentences / day)\(^a\)

\(^a\)with NVIDIA Titan X and amuNMT (https://github.com/emjotde/amunmt)
## Open-vocabulary translation

### Why is vocabulary size a problem?
- Size of one-hot input/output vector is linear to vocabulary size
- Large vocabularies are space inefficient
- Large output vocabularies are time inefficient
- Typical network vocabulary size: 30,000–100,000

### What about out-of-vocabulary words?
- Training set vocabulary typically larger than network vocabulary (1 million words or more)
- At translation time, we regularly encounter novel words:
  - Names: *Barack Obama*
  - Morph. complex words: *Hand/gepäck/gebühr* (‘carry-on bag fee’)
  - Numbers, URLs etc.
Open-vocabulary translation

**Solutions**

- copy unknown words, or translate with back-off dictionary [Jean et al., 2015, Luong et al., 2015b, Gulcehre et al., 2016] → works for names (if alphabet is shared), and 1-to-1 aligned words
- use subword units (characters or others) for input/output vocabulary → model can learn translation of seen words on subword level → model can translate unseen words if translation is *transparent*
- active research area [Sennrich et al., 2016c, Luong and Manning, 2016, Chung et al., 2016, Ling et al., 2015, Costa-jussà and Fonollosa, 2016]
Core idea: transparent translations

transparent translations

- some translations are semantically/phonologically transparent
- morphologically complex words (e.g. compounds):
  - solar system (English)
  - Sonnen|system (German)
  - Nap|rendszer (Hungarian)
- named entities:
  - Obama (English; German)
  - Обама (Russian)
  - オバマ (o-ba-ma) (Japanese)
- cognates and loanwords:
  - claustrophobia (English)
  - Klaustrophobie (German)
  - Клаустрофобия (Russian)
Byte pair encoding [Gage, 1994]

Algorithm

Iteratively replace most frequent byte pair in sequence with unused byte

aaabdaaabac
Byte pair encoding [Gage, 1994]

Algorithm

Iteratively replace most frequent byte pair in sequence with unused byte

\[
\text{aaabdaaabac} \\
\text{ZabdZabac} \quad Z = \text{aa}
\]
Byte pair encoding [Gage, 1994]

**algorithm**

iteratively replace most frequent byte pair in sequence with unused byte

```
aaabdaaabac
ZabdZabac
ZYdZYac
```

```
Z=aa
Y=ab
```
Byte pair encoding [Gage, 1994]

Algorithm

Iteratively replace most frequent byte pair in sequence with unused byte

```
aaabdaaabajac
ZabdZabac
ZYdZYac
XdXac
```

```
Z=aa
Y=ab
X=ZY
```
Byte pair encoding for word segmentation

bottom-up character merging
- iteratively replace most frequent pair of symbols (’A’,’B’) with ’AB’
- apply on dictionary, not on full text (for efficiency)
- output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>’l o w’</td>
<td>5</td>
</tr>
<tr>
<td>’l o w e r’</td>
<td>2</td>
</tr>
<tr>
<td>’n e w e s t’</td>
<td>6</td>
</tr>
<tr>
<td>’w i d e s t’</td>
<td>3</td>
</tr>
</tbody>
</table>
**Byte pair encoding for word segmentation**

**bottom-up character merging**
- iteratively replace most frequent pair of symbols (’A’,’B’) with ’AB’
- apply on dictionary, not on full text (for efficiency)
- output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
<th>(’e’, ’s’)</th>
<th>→</th>
<th>’es’</th>
</tr>
</thead>
<tbody>
<tr>
<td>’l o w &lt;/w&gt;’</td>
<td>5</td>
<td>(’e’, ’s’)</td>
<td>→</td>
<td>’es’</td>
</tr>
<tr>
<td>’l o w e r &lt;/w&gt;’</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>’n e w es t &lt;/w&gt;’</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>’w i d es t &lt;/w&gt;’</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Byte pair encoding for word segmentation

Bottom-up character merging

- iteratively replace most frequent pair of symbols (’A’,’B’) with ’AB’
- apply on dictionary, not on full text (for efficiency)
- output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
<th>(’e’, ’s’)</th>
<th>→</th>
<th>’es’</th>
</tr>
</thead>
<tbody>
<tr>
<td>‘low &lt;/w&gt;’</td>
<td>5</td>
<td>(’es’, ’t’)</td>
<td>→</td>
<td>’est’</td>
</tr>
<tr>
<td>‘lower &lt;/w&gt;’</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>‘newest &lt;/w&gt;’</td>
<td>6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>‘widest &lt;/w&gt;’</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Byte pair encoding for word segmentation

- **bottom-up character merging**
  - iteratively replace most frequent pair of symbols (’A’,’B’) with ’AB’
  - apply on dictionary, not on full text (for efficiency)
  - output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
<th>pair</th>
<th>result</th>
</tr>
</thead>
<tbody>
<tr>
<td>’l o w &lt;/w&gt;’</td>
<td>5</td>
<td>(’e’, ’s’)</td>
<td>’es’</td>
</tr>
<tr>
<td>’l o w e r &lt;/w&gt;’</td>
<td>2</td>
<td>(’es’, ’t’)</td>
<td>’est’</td>
</tr>
<tr>
<td>’n e w est&lt;/w&gt;’</td>
<td>6</td>
<td>(’est’, ’&lt;/w&gt;’)</td>
<td>’est&lt;/w&gt;’</td>
</tr>
<tr>
<td>’w i d est&lt;/w&gt;’</td>
<td>3</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Byte pair encoding for word segmentation

**bottom-up character merging**

- iteratively replace most frequent pair of symbols (‘A’,’B’) with ’AB’
- apply on dictionary, not on full text (for efficiency)
- output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
<th>replace</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>’lo w &lt;/w&gt;’</td>
<td>5</td>
<td>(’e’, ’s’)</td>
<td>’es’</td>
</tr>
<tr>
<td>’lo w e r &lt;/w&gt;’</td>
<td>2</td>
<td>(’es’, ’t’)</td>
<td>’est’</td>
</tr>
<tr>
<td>’n e w est&lt;/w&gt;’</td>
<td>6</td>
<td>(’est’, ’&lt;/w&gt;’)</td>
<td>’est&lt;/w&gt;’</td>
</tr>
<tr>
<td>’w i d est&lt;/w&gt;’</td>
<td>3</td>
<td>(’l’, ’o’)</td>
<td>’lo’</td>
</tr>
</tbody>
</table>
Byte pair encoding for word segmentation

**bottom-up character merging**

- iteratively replace most frequent pair of symbols (‘A’, ‘B’) with ’AB’
- apply on dictionary, not on full text (for efficiency)
- output vocabulary: character vocabulary + one symbol per merge

<table>
<thead>
<tr>
<th>word</th>
<th>frequency</th>
<th>most frequent pair</th>
<th>→</th>
</tr>
</thead>
<tbody>
<tr>
<td>'low &lt;/w&gt;'</td>
<td>5</td>
<td>(‘e’, ‘s’)</td>
<td>'es'</td>
</tr>
<tr>
<td>'low e r &lt;/w&gt;'</td>
<td>2</td>
<td>(‘es’, ‘t’)</td>
<td>'est'</td>
</tr>
<tr>
<td>'n e w est&lt;/w&gt;'</td>
<td>6</td>
<td>(‘est’, '&lt;/w&gt;')</td>
<td>'est&lt;/w&gt;'</td>
</tr>
<tr>
<td>'w i d est&lt;/w&gt;'</td>
<td>3</td>
<td>(‘lo’, ‘w’)</td>
<td>'low'</td>
</tr>
</tbody>
</table>

...
Byte pair encoding for word segmentation

why BPE?

- don’t waste time on frequent character sequences → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'lowest</w>'

- ('e', 's') → 'es'
- ('es', 't') → 'est'
- ('est', '</w>') → 'est</w>'
- ('l', 'o') → 'lo'
- ('lo', 'w') → 'low'
Byte pair encoding for word segmentation

why BPE?
- don’t waste time on frequent character sequences → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'l o w es t </w>'

('e', 's') → 'es'
('es', 't') → 'est'
('est', '</w>') → 'est</w>'
('l', 'o') → 'lo'
('lo', 'w') → 'low'
why BPE?

- don’t waste time on frequent character sequences
  → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'lowest </w>'

('e', 's') → 'es'
('es', 't') → 'est'
('est', '</w>') → 'est</w>'
('l', 'o') → 'lo'
('lo', 'w') → 'low'
Byte pair encoding for word segmentation

why BPE?

- don’t waste time on frequent character sequences → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'lo w est</w>'

('e', 's') → 'es'
('es', 't') → 'est'
('est', '</w>') → 'est</w>'
('l', 'o') → 'lo'
('lo', 'w') → 'low'
why BPE?

- don’t waste time on frequent character sequences
  → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'lo w est</w>'

- ('e', 's') → 'es'
- ('es', 't') → 'est'
- ('est', '</w>') → 'est</w>'
- ('l', 'o') → 'lo'
- ('lo', 'w') → 'low'
why BPE?

- don’t waste time on frequent character sequences
  → trade-off between text length and vocabulary sizes
- open-vocabulary:
  learned operations can be applied to unknown words
- alternative view: character-level model on compressed text

'low est</w>’

('e', 's') → 'es'
('es', 't') → 'est'
('est', '</w>') → 'est</w>'
('l', 'o') → 'lo'
('lo', 'w') → 'low'
motivation: disambiguate words by POS

<table>
<thead>
<tr>
<th>English</th>
<th>German</th>
</tr>
</thead>
<tbody>
<tr>
<td>close\textsubscript{verb}</td>
<td>schließen</td>
</tr>
<tr>
<td>close\textsubscript{adj}</td>
<td>nah</td>
</tr>
<tr>
<td>close\textsubscript{noun}</td>
<td>Ende</td>
</tr>
</tbody>
</table>

source: We thought a win like this might be close\textsubscript{adj}.

reference: Wir dachten, dass ein solcher Sieg nah sein könnte.

baseline NMT: *Wir dachten, ein Sieg wie dieser könnte schließen.
Linguistic Features: Architecture

use separate embeddings for each feature, then concatenate

baseline: only word feature

\[ E(\text{close}) = \begin{bmatrix} 0.5 \\ 0.2 \\ 0.3 \\ 0.1 \end{bmatrix} \]

<table>
<thead>
<tr>
<th>( F )</th>
<th>input features</th>
</tr>
</thead>
<tbody>
<tr>
<td>( E_1(\text{close}) = \begin{bmatrix} 0.4 \ 0.1 \ 0.2 \end{bmatrix} )</td>
<td>( E_2(\text{adj}) = \begin{bmatrix} 0.1 \end{bmatrix} )</td>
</tr>
</tbody>
</table>
Linguistic Features: Results

experimental setup

- WMT 2016 (parallel data only)
- source-side features:
  - POS tag
  - dependency label
  - lemma
  - morphological features
  - subword tag
Architecture variants

an incomplete selection

- convolutional network as encoder [Kalchbrenner and Blunsom, 2013]
- TreeLSTM as encoder [Eriguchi et al., 2016]
- modifications to attention mechanism [Luong et al., 2015a, Feng et al., 2016]
- deeper networks [Zhou et al., 2016]
- coverage model [Mi et al., 2016, Tu et al., 2016b, Tu et al., 2016a]
- reward symmetry between source-to-target and target-to-source attention [Cohn et al., 2016, Cheng et al., 2015]
Sequence-level training

- problem: at training time, target-side history is reliable; at test time, it is not.
  → exposure bias

- solution: instead of using gold context, sample from the model to obtain target context
  [Shen et al., 2016, Ranzato et al., 2016, Bengio et al., 2015]

- more efficient cross entropy training remains in use to initialize weights
### Trading-off target and source context

<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>uedin-nmt</td>
<td>A year later, Fed officials reversed those cuts.</td>
</tr>
<tr>
<td>uedin-pbsmt</td>
<td>A year later, FedEx officials reversed those cuts.</td>
</tr>
<tr>
<td></td>
<td>A year later, the Fed representatives made these cuts.</td>
</tr>
</tbody>
</table>

### Problem

- RNN is locally normalized at each time step

  - given *Fed*: as previous (sub)word, *Ex* is very likely in training data:

    $p(Ex|Fed:) = 0.55$

- *label bias problem*: locally-normalized models may ignore input in low-entropy state

### Potential solutions (speculative)

- sampling at training time

- bidirectional decoder [Liu et al., 2016, Sennrich et al., 2016a]

- context gates to trade-off source and target context [Tu et al., 2016]
Training data: monolingual

Why train on monolingual data?
- cheaper to create/collect
- parallel data is scarce for many language pairs
- domain adaptation with *in-domain* monolingual data
Training data: monolingual

Solutions/1 [Gülçehre et al., 2015]

shallow fusion: rescore beam with language model
deepe fusion: extra, LM-specific hidden layer

Figure 1: Graphical illustrations of the proposed fusion methods.

learned by the LM from monolingual corpora is not overwritten. It is possible to use monolingual corpora as well while finetuning all the parameters, but in this paper, we alter only the output parameters in the stage of finetuning.

4.2.1 Balancing the LM and TM
In order for the decoder to flexibly balance the input from the LM and TM, we augment the decoder with a "controller" mechanism. The need to flexibly balance the signals arises depending on the work being translated. For instance, in the case of Zh-En, there are no Chinese words that correspond to articles in English, in which case the LM may be more informative. On the other hand, if a noun is to be translated, it may be better to ignore any signal from the LM, as it may prevent the decoder from choosing the correct translation. Intuitively, this mechanism helps the model dynamically weight the different models depending on the word being translated.

The controller mechanism is implemented as a function taking the hidden state of the LM as input and computing

\[ g_t = \sigma(v^\top g_{LM} + b_g), \]

where \( \sigma \) is a logistic sigmoid function.

The output of the controller is then multiplied with the hidden state of the LM. This lets the decoder use the signal from the TM fully, while the controller controls the magnitude of the LM signal.

In our experiments, we empirically found that it was better to initialize the bias \( b_g \) to a small, negative number. This allows the decoder to decide the importance of the LM only when it is deemed necessary.

5 Datasets
We evaluate the proposed approaches on four diverse tasks: Chinese to English (Zh-En), Turkish to English (Tr-En), German to English (De-En) and Czech to English (Cs-En). We describe each of these datasets in more detail below.

5.1 Parallel Corpora
5.1.1 Zh-En: OpenMT’15
We use the parallel corpora made available as a part of the NIST OpenMT’15 Challenge. Sentence-aligned pairs from three domains are combined to form a training set: (1) SMS/CHAT and (2) conversational telephone speech (CTS) from DARPA BOLT Project, and (3) newsgroups/weblogs from DARPA GALE Project. In total, the training set consists of 430K sentence pairs (see Table 1 for the detailed statistics). We train...
Training data: monolingual

Solutions/2 [Sennrich et al., 2016b]

- decoder is already a language model
  → mix monolingual data into training set
- problem: how to get $c_i$ for monolingual training instances?
  - dummy source context $c_i$ (moderately effective)
  - produce synthetic source sentence via back-translation
    → get approximation of $c_i$
Multi-source translation [Zoph and Knight, 2016]

we can condition on multiple input sentences

- benefits:
  - one source text may contain information that is unspecified in other → possible quality gains

- drawbacks:
  - we need multiple source sentences at training and decoding time

Training data: multilingual
Multilingual models [Dong et al., 2015, Firat et al., 2016]
we can share layers of the model across language pairs

Figure 2: Multi-task learning framework for multiple-target language translation

Figure 3: Optimization for end to multi-end model

3.4 Translation with Beam Search

Although parallel corpora are available for the encoder and the decoder modeling in the training phrase, the ground truth is not available during test time. During test time, translation is produced by finding the most likely sequence via beam search.

\[
\hat{Y} = \arg\max_Y \ p(Y^T | S^T)
\]

Given the target direction we want to translate to, beam search is performed with the shared encoder and a specific target decoder where search space belongs to the decoder \(T_p\). We adopt beam search algorithm similar as it is used in SMT system (Koehn, 2004) except that we only utilize scores produced by each decoder as features. The size of beam is 10 in our experiments for speedup consideration. Beam search is ended until the end-of-sentence \(\text{eos}\) symbol is generated.

4 Experiments

We conducted two groups of experiments to show the effectiveness of our framework. The goal of the first experiment is to show that multi-task learning helps to improve translation performance given enough training corpora for all language pairs. In the second experiment, we show that for some resource-poor language pairs with a few parallel training data, their translation performance could be improved as well.

4.1 Dataset

The Europarl corpus is a multi-lingual corpus including 21 European languages. Here we only choose four language pairs for our experiments. The source language is English for all language pairs. And the target languages are Spanish (Es), French (Fr), Portuguese (Pt) and Dutch (Nl). To demonstrate the validity of our learning framework, we do some preprocessing on the training set. For the source language, we use 30k of the most frequent words for source language vocabulary which is shared across different language pairs and 30k most frequent words for each target language. Out-of-vocabulary words are denoted as unknown words, and we maintain different unknown word labels for different languages. For test sets, we also restrict all words in the test set to be from our training vocabulary and mark the OOV words as the corresponding labels as in the training data. The size of training corpus in experiment 1 and 2 is listed in Table 1 where

---

**benefits:**

- transfer learning from one language pair to the other → possible quality gains, especially for low-resourced language pairs
- scalability: do we need \(N^2 - N\) independent models for \(N\) languages? → sharing of parameters allows linear growth → zero-shot translation?
Training data: other tasks

Multi-task models [Luong et al., 2016]

- other tasks can be modelled with sequence-to-sequence models
- we can share layers between translation and other tasks

Figure 1: Sequence to sequence learning examples – (left) machine translation (Sutskever et al., 2014) and (right) constituent parsing (Vinyals et al., 2015a).

Furthermore, we have established a new state-of-the-art result in constituent parsing with 93.0 F1.

We also explore two unsupervised learning objectives, sequence autoencoders (Dai & Le, 2015) and skip-thought vectors (Kiros et al., 2015), and reveal their interesting properties in the MTL setting: autoencoder helps less in terms of perplexities but more on BLEU scores compared to skip-thought.

Sequence to sequence learning (seq2seq) aims to directly model the conditional probability \( p(y|x) \) of mapping an input sequence, \( x_1, \ldots, x_n \), into an output sequence, \( y_1, \ldots, y_m \). It accomplishes such goal through the encoder-decoder framework proposed by Sutskever et al. (2014) and Cho et al. (2014). As illustrated in Figure 1, the encoder computes a representation \( s \) for each input sequence. Based on that input representation, the decoder generates an output sequence, one unit at a time, and hence, decomposes the conditional probability as:

\[
\log p(y|x) = \sum_{j=1}^{m} \log p(y_j|y_{<j}, x, s)
\]  

A natural model for sequential data is the recurrent neural network (RNN), which is used by most of the recent seq2seq work. These work, however, differ in terms of: (a) architecture – from unidirectional, to bidirectional, and deep multi-layer RNNs; and (b) RNN type – which are long-short term memory (LSTM) (Hochreiter & Schmidhuber, 1997) and the gated recurrent unit (Cho et al., 2014).

Another important difference between seq2seq work lies in what constitutes the input representation \( s \). The early seq2seq work (Sutskever et al., 2014; Cho et al., 2014; Luong et al., 2015b; Vinyals et al., 2015b) uses only the last encoder state to initialize the decoder and sets \( s = [\] \) in Eq. (1). Recently, Bahdanau et al. (2015) proposes an attention mechanism, a way to provide seq2seq models with a random access memory, to handle long input sequences. This is accomplished by setting \( s \) in Eq. (1) to be the set of encoder hidden states already computed. On the decoder side, at each time step, the attention mechanism will decide how much information to retrieve from that memory by learning where to focus, i.e., computing the alignment weights for all input positions.

Recent work such as (Xu et al., 2015; Jean et al., 2015a; Luong et al., 2015a; Vinyals et al., 2015a) has found that it is crucial to empower seq2seq models with the attention mechanism.

We generalize the work of Dong et al. (2015) to the multi-task sequence-to-sequence learning setting that includes the tasks of machine translation (MT), constituency parsing, and image caption generation. Depending which tasks involved, we propose to categorize multi-task seq2seq learning into three general settings. In addition, we will discuss the unsupervised learning tasks considered as well as the learning process.

3.1 One-to-many setting

This scheme involves one encoder and multiple decoders for tasks in which the encoder can be shared, as illustrated in Figure 2. The input to each task is a sequence of English words. A separate decoder is used to generate each sequence of output units which can be either (a) a sequence of tags.
NMT as a component in log-linear models

Log-linear models

- model ensembling is well-established
- reranking output of phrase-based/syntax-based with NMT [Neubig et al., 2015]
- incorporating NMT as a feature function into PBSMT [Junczys-Dowmunt et al., 2016]
  → results depend on relative performance of PBSMT and NMT
- log-linear combination of different neural models
  - left-to-right and right-to-left [Liu et al., 2016]
  - source-to-target and target-to-source [Li and Jurafsky, 2016]
Some future directions for (neural) MT research

- (better) solutions to new(ish) problems
  - OOVs, coverage, efficiency...
- work on "hard" translation problems
  - consider context beyond sentence boundary
  - reward semantic adequacy of translation
  - ...
- new opportunities
  - one model for many language pairs?
  - tight integration with other NLP tasks
Further Reading

- lecture notes by Kyunghyun Cho: [Cho, 2015]
(A small selection of) Resources

NMT tools

- dl4mt-tutorial (theano) [https://github.com/nyu-dl/dl4mt-tutorial](https://github.com/nyu-dl/dl4mt-tutorial)
  (our branch: nematus [https://github.com/rsennrich/nematus](https://github.com/rsennrich/nematus))
- nmt.matlab [https://github.com/lmthang/nmt.matlab](https://github.com/lmthang/nmt.matlab)
- seq2seq (tensorflow) [https://www.tensorflow.org/versions/r0.8/tutorials/seq2seq/index.html](https://www.tensorflow.org/versions/r0.8/tutorials/seq2seq/index.html)
- neural monkey (tensorflow) [https://github.com/ufal/neuralmonkey](https://github.com/ufal/neuralmonkey)
Do it yourself

- Sample files and instructions for training NMT model
  https://github.com/rsennrich/wmt16-scripts
- Pre-trained models to test decoding (and for further experiments)
  http://statmt.org/rsennrich/wmt16_systems/

Lab session this afternoon

- Install Nematus
- Use Nematus with existing model
- Adapt existing model to new domain via continued training
Thank you!
Neural Machine Translation by Jointly Learning to Align and Translate.  
In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks.  
*CoRR*, abs/1506.03099.

Neural versus Phrase-Based Machine Translation Quality: a Case Study.  
In *EMNLP 2016*.


Agreement-based Joint Training for Bidirectional Attention-based Neural Machine Translation.  
*CoRR*, abs/1512.04650.

Natural Language Understanding with Distributed Representation.  
*CoRR*, abs/1511.07916.

Describing Multimedia Content using Attention-based Encoder-Decoder Networks.


Li, J. and Jurafsky, D. (2016).
Mutual Information and Diverse Decoding Improve Neural Machine Translation.
CoRR, abs/1601.00372.

Character-based Neural Machine Translation.
ArXiv e-prints.

Agreement on Target-bidirectional Neural Machine Translation.
In NAACL HLT 16, San Diego, CA.

Multi-task Sequence to Sequence Learning.
In ICLR 2016.


Luong, M.-T. and Manning, D. C. (2016).
Achieving Open Vocabulary Neural Machine Translation with Hybrid Word-Character Models.

Luong, T., Pham, H., and Manning, C. D. (2015a).
Effective Approaches to Attention-based Neural Machine Translation.
Addressing the Rare Word Problem in Neural Machine Translation.

ArXiv e-prints.

Neural Reranking Improves Subjective Quality of Machine Translation: NAIST at WAT2015.

Sequence Level Training with Recurrent Neural Networks.
In ICLR 2016.

Linguistic Input Features Improve Neural Machine Translation.

Edinburgh Neural Machine Translation Systems for WMT 16.
Improving Neural Machine Translation Models with Monolingual Data.

Neural Machine Translation of Rare Words with Subword Units.

In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Berlin, Germany.

Sequence to Sequence Learning with Neural Networks.

ArXiv e-prints.

Coverage-based Neural Machine Translation.
CoRR, abs/1601.04811.
Modeling Coverage for Neural Machine Translation.


Multi-Source Neural Translation.
In NAACL HLT 2016.