Introduction to (Neural) Machine Translation

Ondřej Bojar

September 5, 2022
Outline

• Statistical Machine Translation
  • Why “classical” SMT fails

• Neural Machine Translation
  • Learning Representations
  • Processing Text
  • Transformer (Self-Attention)

• Some of Training Magic

• Caveats on Interpreting Results

Some slides by Jindřich Helcl, Jindřich Libovický, Tom Kocmi. Many slides based on slides by Rico Sennrich and others.
Further Reading

Slides from my subject here at ÚFAL:

https://ufal.mff.cuni.cz/courses/npf1087

Videolectures & Wiki (everything before neural):

http://mttalks.ufal.ms.mff.cuni.cz/

Books and others:


  Chapter on NMT: https://arxiv.org/abs/1709.07809

• Ondřej Bojar: Čeština a strojový překlad. ÚFAL, 2012.
• The deeper analysis, the easier the **transfer** should be.
• A hypothetical interlingua captures pure meaning.
• Statistical systems learn “automatically” from data.
• Rule-based systems implemented by linguists-programmers.

Until NMT, it was best to combine the approaches.
The Statistical Approach

(Statistical = Information-theoretic.)

• Specify a probabilistic model.
  = How is the probability mass distributed among possible outputs given observed inputs.

• Specify the training criterion and procedure.
  = How to learn free parameters from training data.

Notice:

• Linguistics helpful when designing the models:
  • How to divide input into smaller units.
  • Which bits of observations are more informative.
Statistical MT

Given a source (foreign) language sentence $f_1^J = f_1 \ldots f_j \ldots f_J$, produce a target language (English) sentence $e_1^I = e_1 \ldots e_j \ldots e_I$. Among all possible target language sentences, choose the sentence with the highest probability:

$$\hat{e}_1^I = \arg\max_{I, e_1^I} p(e_1^I | f_1^J) \quad (1)$$

We stick to the $e_1^I, f_1^J$ notation regardless the source and target languages.
Brute-Force MT

Translate only sentences listed in a “translation memory” (TM):

- Good morning. = Dobré ráno.
- How are you? = Jak se máš?
- How are you? = Jak se máte?

\[
p(e_1^I | f_1^J) = \begin{cases} 
1 & \text{if } e_1^I = f_1^J \text{ seen in the TM} \\
0 & \text{otherwise} 
\end{cases} \tag{2}
\]

Any problems with the definition?
Brute-Force MT

Translate only sentences listed in a “translation memory” (TM):

Good morning. = Dobré ráno.
How are you? = Jak se máš?
How are you? = Jak se máte?

\[
p(e^I_1 | f^J_1) = \begin{cases} 
1 & \text{if } e^I_1 = f^J_1 \text{ seen in the TM} \\
0 & \text{otherwise} 
\end{cases}
\]

• Not a probability. There may be \(f^J_1\), s.t. \(\sum e^I_1 p(e^I_1 | f^J_1) > 1\).

⇒ Have to normalize, use \(\frac{\text{count}(e^I_1, f^J_1)}{\text{count}(f^J_1)}\) instead of 1.

• Not “smooth”, no generalization:

  Good morning. ⇒ Dobré ráno.
  Good evening. ⇒ ∅
Old School: Noisy Channel Model

\[ \hat{e}_1^I = \arg\max_{I,e_1^I} p(e_1^I|f_1^J) = \arg\max_{I,e_1^I} p(f_1^J|e_1^I)p(e_1^I) \] (3)

Bayes’ law divided the model into components:

\[ p(f_1^J|e_1^I) \] Translation model ("reversed", \( e_1^I \rightarrow f_1^J \))
   ...is it a likely translation?

\[ p(e_1^I) \] Language model (LM)
   ...is the output a likely sentence of the target language?

- The components can be trained on different sources.
  There are far more monolingual data \( \Rightarrow \) language model can be more reliable.

... smoothing enabled by a math trick.
Without Equations

Input Global Search
for sentence with highest probability

Parallel Texts

Translation Model

Monolingual Texts

Language Model

Output
Translation Model of Phrase-Based MT

- The key element was phrase translation probability:
  \[ p(\tilde{f}_k|\tilde{e}_k) = \frac{\text{count}(\tilde{f}, \tilde{e})}{\text{count}(\tilde{e})} \quad \text{... how often } \tilde{f}_k \text{ served as the translation of } \tilde{e}_k. \]

- The whole source sentence was covered with phrase translations.

- The total translation probability was the product of individual phrase translation probabilities:
  \[ h_{\text{Phr}}(f^J_1, e^I_1, s^K_1) = \log \prod_{k=1}^{K} p(\tilde{f}_k|\tilde{e}_k) \quad \text{... smoothing via decomposition into smaller units.} \]
Work on Your Data

- CzEng (Czech-English) reached 180M million sentence pairs:
  - 0.6 cs / 0.7 en gigawords of genuine parallel text (61M sentpairs)
  - 2.0 cs / 2.3 en gigawords of synthetic text (127M sentpairs)

<table>
<thead>
<tr>
<th>Ver.</th>
<th>S. Pairs</th>
<th>Main Focus</th>
<th>Details in</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.5</td>
<td>0.9M</td>
<td>Sentence alignment, common format</td>
<td>Bojar and Žabokrtský (2006)</td>
</tr>
<tr>
<td>0.7</td>
<td>1.0M</td>
<td>Used in WMT06 and WMT07</td>
<td>Bojar et al. (2008)</td>
</tr>
<tr>
<td>0.9</td>
<td>8.0M</td>
<td>Automatic annotation up to t-layer</td>
<td>Bojar and Žabokrtský (2009)</td>
</tr>
<tr>
<td>–</td>
<td>–</td>
<td>Sentence-level filtering</td>
<td>Bojar et al. (2010)</td>
</tr>
<tr>
<td>1.0</td>
<td>15.0M</td>
<td>Improving monolingual annotation through parallel data</td>
<td>Bojar et al. (2012)</td>
</tr>
<tr>
<td>1.6</td>
<td>62.5M</td>
<td>Processing tools dockered</td>
<td>Bojar et al. (2016)</td>
</tr>
<tr>
<td>1.7</td>
<td>57.1M</td>
<td>Block-level filtering</td>
<td>–</td>
</tr>
<tr>
<td>2.0</td>
<td>188.0M</td>
<td>Filtering + Synthetic data</td>
<td>–</td>
</tr>
</tbody>
</table>
In my dream, there was a sycamore growing out of the ruins of the sacristy, and I was told that, if I dug at the roots of the sycamore, I would find a hidden treasure. But I'm not so stupid as to cross an entire desert just because of a recurrent dream. "And they disappeared. The boy stood up shakily, and looked once more at the Pyramids. "It is I who dared to do so," said the boy. This man looked exactly the same, except that now the roles were reversed. "It is I who dared to do so," he
In my dream, there was a sycamore growing out of the ruins of the sacristy, and I was told that, if I dug at the roots of the sycamore, I would find a hidden treasure.

But I’m not so stupid as to cross an entire desert just because of a recurrent dream.

And they disappeared.

The boy stood up shakily, and looked once more at the Pyramids.

"It is I who dared to do so," said the boy.

This man looked exactly the same, except that now the roles were reversed.

"It is I who dared to do so," he repented, and lowered his head to receive a blow from the sword.

"Life was good to me," the man said.

"When you appeared in my dream, I felt that all my efforts had been rewarded, because my son’s poems will be read by men for generations to come.

But any father would be proud of the fame achieved by one whom he had cared for as a child, and educated as he grew up.

"We’re two very different things," the man said.

"That’s not true," the boy said.

I learned the alchemist’s secrets in my travels.

I have inside me the winds, the deserts, the oceans, the stars, and everything created in the universe.

We were all made by the same hand, and we have the same soul.

You’ll learn to love the desert, and you’ll get to know every one of the fifty thousand palms.

You’ll watch them as they grow, demonstrating how the world is always changing.

And you’ll get better and better at understanding men, because the desert is the best teacher there is.

"Sometimes during the second year... you’ll remember about the treasure.

The omen will begin insensibly to speak of it, and you’ll try to ignore them.

But you know that I’m not going to go to Mecca. Just as you know that you’re not going to buy your sheep.

"Who told you that?" asked the boy, startled.

"Makthub" said the old crystal merchant.

And he gave the boy his blessing.

The boy went to his room and packed his belongings.

They filled three sacks.

As he was leaving, he saw, in the corner of the room, his old shepherd’s satchel.

"I want to see the greatness of Allah," the chief said, with respect.

"I want to see how a man turns himself into the wind.

But he made a mental note of the names of the two men who had expressed their fear.

Gale and Church (1993) illustrated in MT Talk #7 (https://youtu.be/_4lnyoC3mtQ)
From Sent. Pairs, Learn Word Translations

<table>
<thead>
<tr>
<th>bílý</th>
<th>dům</th>
<th>white</th>
<th>house</th>
<th>white</th>
<th>house</th>
<th>white</th>
<th>house</th>
</tr>
</thead>
<tbody>
<tr>
<td>bílý</td>
<td>pes</td>
<td>white</td>
<td>dog</td>
<td>white</td>
<td>dog</td>
<td>white</td>
<td>dog</td>
</tr>
<tr>
<td>bílý</td>
<td>pes</td>
<td>black</td>
<td>dog</td>
<td>black</td>
<td>dog</td>
<td>black</td>
<td>dog</td>
</tr>
<tr>
<td>černý</td>
<td>pes</td>
<td>black</td>
<td>dog</td>
<td>black</td>
<td>dog</td>
<td>black</td>
<td>dog</td>
</tr>
</tbody>
</table>

IBM Model 1 illustrated in Bojar (2012). See also MT Talk #8 (https://youtu.be/mqyMDLu5JPw)
Nemám žádného psa. I have no dog.
Viděl kočku. He saw a cat.
Nemám žádného psa.
I have no dog.
Viděl kočku.
He saw a cat.
Nemám žádného psa.  
I have no dog.  
Viděl kočku.  
He saw a cat.
New input: Nemám kočku.
Nemám žádného psa. I have no dog. Viděl kočku. He saw a cat.

New input: Nemám kočku. ... I don't have cat.
Nemám žádného psa.
I have no dog.
Viděl kočku.
a cat. He saw

New input: Nemám kočku.
... I don't have cat.
6: So That $n$-Grams Probable (LM)

New input: I have no dog.

Nemám žádného psa.

Viděl kočku.

... I don't have cat.

New input: I have a cat.
Meaning Got Reversed!

Nemám žádného psa. (I have no dog.)

Viděl kočku. (He saw a cat.)

New input: Nemám kočku. (I don't have a cat.)

...I don't have cat.
What Went Wrong?

\[ \hat{e}_1^I = \arg \max_{I, e_1^I} p(f_1^J | e_1^I)p(e_1^I) = \arg \max_{I, e_1^I} \prod_{(\hat{f}, \hat{e}) \in \text{phrase pairs of } f_1^J, e_1^I} p(\hat{f} | \hat{e})p(e_1^I) \] (4)

Too strong independence assumptions:

- Language model as a separate unit.
  - \( p(e_1^I) \) models the target sentence independently of \( f_1^J \).
- Phrases translated independent of one another.
  - In fact, phrases do depend on each other. Here “nemám” and “žádného” jointly express one negation.
  - Word alignments ignored that dependence. But adding it would increase data sparseness.
  - LM was the only means for glueing phrases, but it prefers positive sents.
Redefining $p(e^I_1|f^J_1)$

What if we modelled $p(e^I_1|f^J_1)$ directly, word by word:

$$p(e^I_1|f^J_1) = p(e_1, e_2, \ldots, e_I|f^J_1)$$
$$= p(e_1|f^J_1) \cdot p(e_2|e_1, f^J_1) \cdot p(e_3|e_2, e_1, f^J_1) \cdots$$
$$= \prod_{i=1}^{I} p(e_i|e_1, \ldots, e_{i-1}, f^J_1)$$

(5)

...this is “just a cleverer language model:”

$$p(e^I_1) = \prod_{i=1}^{I} p(e_i|e_1, \ldots, e_{i-1})$$

Main Benefit: All dependencies available.

But what technical device can learn this?
• A neural network with a single hidden layer (possibly huge) can approximate any continuous function to any precision.
• (Nothing claimed about learnability in practice.)

Play with playground.tensorflow.org

\[-0.43x_1 - 0.89x_2 + 2.0 > 0\]
and 
\[-0.67x_1 + 0.89x_2 + 2.1 > 0\]
and 
\[1.4x_1 - 0.067x_2 + 2.3 > 0\]
A DL “Program” Is Just a Computation...

In fact: \(1 \tanh(-0.43x_1 - 0.89x_2 + 2.0) + 1 \tanh(-0.67x_1 + 0.89x_2 + 2.1) + 1 \tanh(1.4x_1 - 0.067x_2 + 2.3) - \pi/2 > 0\)
In fact:  

\[
1 \tanh(-0.43x_1 - 0.89x_2 + 2.0) \\
+ 1 \tanh(-0.67x_1 + 0.89x_2 + 2.1) \\
+ 1 \tanh(1.4x_1 - 0.067x_2 + 2.3) - \pi/2 > 0
\]
Perfect Features

\[ 1x_1^2 + 1x_2^2 - 1 < 0 \]
Bad Features & Low Depth
Too Complex NN Fails to Learn
One Fully Connected Layer

- One fully-connected layer converts an input (column) vector $x$ to an output (column) vector $h$:

$$h = f(Wx + b),$$  

(6)

- $W$ is a weight matrix of input columns and output rows,
- $b$ a bias vector of length of output,
- $f(\cdot)$ is a non-linearity applied usually elementwise.

![Diagram of a fully connected layer with input $x$, weight matrix $W$, bias $b$, and non-linearity $f$ applied to the sum $Wx + b$.]
One Layer $\tanh(Wx + b)$, 2D $\rightarrow$ 2D

Skew: $W$

Transpose: $b$

Non-lin.: $\tanh$

Feed-Forward Neural Network

\[ x \]
\[ h_1 = f(W_1 x + b_1) \]
\[ h_2 = f(W_2 h_1 + b_2) \]
\[ \vdots \]
\[ h_n = f(W_n h_{n-1} + b_n) \]
\[ o = g(W_o h_n + b_o) \]
\[ E = e(o, t) \]

\[ \frac{\partial E}{\partial W_o} = \frac{\partial E}{\partial o} \cdot \frac{\partial o}{\partial W_o} \]

blue: Training item (input \( x \), output \( t \)), red: Trainable parameters \((W_1, b_1, \ldots)\).
Four Layers, Disentangling Spirals

Deep NNs for Image Classification

It's deep if it has more than one stage of non-linear feature transformation.
Processing Text with NNs

• Map each word to a vector of 0s and 1s ("1-hot repr."): 
  
  \[ \text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0) \]

• Sentence is then a matrix:

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>cat</th>
<th>is</th>
<th>on</th>
<th>the</th>
<th>mat</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>about</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>cat</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>is</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>the</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>zebra</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Main drawback: No relations, all words equally close/far.
### Processing Text with NNs

- **Map each word to a vector of 0s and 1s (“1-hot repr.”):**
  
  \[ \text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0) \]

- **Sentence is then a matrix:**

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>cat</th>
<th>is</th>
<th>on</th>
<th>the</th>
<th>mat</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>about</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>cat</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>is</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>the</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>zebra</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

**Vocabulary size:**
- 1.3M English
- 2.2M Czech

Main drawback: No relations, all words equally close/far.
• Map each word to a vector of 0s and 1s ("1-hot repr."):  
  \[ \text{cat} \mapsto (0, 0, \ldots, 0, 1, 0, \ldots, 0) \]

• Sentence is then a matrix:

<table>
<thead>
<tr>
<th></th>
<th>the</th>
<th>cat</th>
<th>is</th>
<th>on</th>
<th>the</th>
<th>mat</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>about</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>...</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cat</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Vocabulary size: 1.3M English, 2.2M Czech

Main drawback: No relations, all words equally close/far. (Smoothing!)
1: Ensuring Similarity: Word Embeddings

• Idea: Map each word to a dense vector.
• Result: 300–2000 dimensions instead of 1–2M.
  • The dimensions have no clear interpretation.
• The “embedding” is the mapping.
  • Technically, the first layer of NNs for NLP is the matrix that maps 1-hot input to the first layer.

• Embeddings are trained for each particular task.
  • Sentence classification (sentiment analysis, etc.)
  • Neural language modelling.
  • The famous word2vec (Mikolov et al., 2013):
    • CBOW: Predict the word from its four neighbours.
    • Skip-gram: Predict likely neighbours given the word.
  • End-to-end neural MT.
2: Reducing Voc. Size: Sub-Words

- SMT struggled with productive morphology (>1M wordforms).
  nejneobhodpodávávatelnějšími, Donaudampfschifffahrtsgesellschaftskapitän
- NMT can handle vocabulary of only only 30–80k entries.

⇒ Resort to sub-word units.

<table>
<thead>
<tr>
<th></th>
<th>český politik svezl migranty</th>
</tr>
</thead>
<tbody>
<tr>
<td>Syllables</td>
<td>čes ký po li tik sve zl mig ran ty</td>
</tr>
<tr>
<td>Morphemes</td>
<td>česk ý politik s vez l migrant y</td>
</tr>
<tr>
<td>Char Pairs</td>
<td>če sk ý po li ti k sv ez l mi gr an ty</td>
</tr>
<tr>
<td>Chars</td>
<td>č e s k ý p o l i t i k s v e z l m i g r a n t y</td>
</tr>
<tr>
<td>BPE 30k</td>
<td>český politik s@@ vez@@ l mi@@ granty</td>
</tr>
</tbody>
</table>

BPE (Byte-Pair Encoding, (Sennrich et al., 2016)) or Google’s wordpieces (Wu et al., 2016) and Tensor2Tensor’s SubwordTextEncoder use \( n \) most common substrings (incl. frequent words).

Variable-length input can be handled by recurrent NNs:

- Processing one input symbol at a time.
  - Initial state \( h_0 = (0) \) (or some sentence representation).
  - The same (trained) transformation \( A \) used every time.

\[
h_t = A(h_{t-1}, x_t)
\]  

- Unroll in time (up to a fixed length limit).
Vanilla RNN

\[ h_t = \tanh(W[h_{t-1}; x_t] + b) \]  \hspace{1cm} (8)

- Vanishing gradient problem.
- Non-linear transformation always applied.
  \[ [h_{t-1}; x_t] \] is concatenation of \( h_{t-1} \) and \( x_t \)
  \Rightarrow Type theory: \( h_t \) and \( h_{t-1} \) live in different vector spaces.
LSTM and GRU Cells for RNN

- LSTM, Long Short-Term Memory Cells (Hochreiter and Schmidhuber, 1997).
- GRU, Gated Recurrent Unit Cells (Chung et al., 2014):

\[
\begin{align*}
    z_t &= \sigma(W_z[h_{t-1}; x_t] + b_z) \quad (9) \\
    r_t &= \sigma(W_r[h_{t-1}; x_t] + b_r) \quad (10) \\
    \tilde{h}_t &= \tanh(W[r_t \odot h_{t-1}; x_t]) \quad (11) \\
    h_t &= (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (12)
\end{align*}
\]

- Gates control:
  - what to use from input \(x_t\) (GRU: everything),
  - what to use from hidden state \(h_{t-1}\) (reset gate \(r_t\)),
  - what to put into output (update gate \(z_t\))

- Linear “information highway” preserved.
  \[ \Rightarrow \text{All states } h_t \text{ belong to the same vector space.} \]
Cho et al. (2014) proposed:

- encoder-decoder architecture and
- GRU unit (name given later by Chung et al. (2014))
- to score variable-length phrase pairs in PBMT.
Embeddings of Phrases
Syntactic Similarity ("of the")
⇒ Semantic Similarity (Countries)
RNN Language Model

• Train RNN as a **classifier for next words** (unlimited history):

  ![Diagram of RNN as a classifier for next words]

• Can be used:
  - To estimate sentence probability / perplexity.
  - To sample from the distribution:

  ![Diagram of sampling from the distribution]
Two Views on RNN LM

- RNN is a for loop / functional map over sequential data
- all outputs are conditional distributions
  \[ P (w_1, \ldots, w_n) = \prod_{i=1}^{n} P(w_i | w_{i-1}, \ldots, w_1) \]
RNN for Translation: Encoder-Decoder

\[
\langle s \rangle \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_4
\]

\[
\sim y_1 \sim y_2 \sim y_3 \sim y_4 \sim y_5
\]
RNN for Translation: Encoder-Decoder

source language input + target language LM
Encoder-Decoder Model – Formal Notation

Data
input tokens (source language) $\mathbf{x} = (x_1, \ldots, x_{T_x})$
output tokens (target language) $\mathbf{y} = (y_1, \ldots, y_{T_y})$
Data
input tokens (source language) \( \mathbf{x} = (x_1, \ldots, x_{T_x}) \)
output tokens (target language) \( \mathbf{y} = (y_1, \ldots, y_{T_y}) \)

Encoder
initial state \( h_0 \equiv 0 \)
\( j \)-th state \( h_j = \text{RNN}_{\text{enc}}(h_{j-1}, x_j) = \tanh(U_e h_{j-1} + W_e E_e x_j + b_e) \)
final state \( h_{T_x} \)

Decoder
initial state \( s_0 = h_{T_x} \)
\( i \)-th decoder state \( s_i = \text{RNN}_{\text{dec}}(s_{i-1}, \hat{y}_{i-1}) = \tanh(U_d s_{i-1} + W_d E_d \hat{y}_{i-1} + b_d) \)
\( i \)-th word score \( t_i = \tanh(U_o s_i + W_o E_d \hat{y}_{i-1} + b_o) \) ("output projection")
output \( \hat{y}_i = \arg \max V_o t_i \)
Encoder-Decoder Model – Formal Notation

Data
input tokens (source language) \( \mathbf{x} = (x_1, \ldots, x_{T_x}) \)
output tokens (target language) \( \mathbf{y} = (y_1, \ldots, y_{T_y}) \)

Encoder
initial state \( h_0 \equiv \mathbf{0} \)
\( j \)-th state \( h_j = \text{RNN}_{\text{enc}}(h_{j-1}, x_j) = \tanh(U_e h_{j-1} + W_e E_e x_j + b_e) \)
final state \( h_{T_x} \)

Decoder
initial state \( s_0 = h_{T_x} \)
\( i \)-th decoder state \( s_i = \text{RNN}_{\text{dec}}(s_{i-1}, \hat{y}_{i-1}) = \tanh(U_d s_{i-1} + W_d E_d \hat{y}_{i-1} + b_d) \)
\( i \)-th word score \( t_i = \tanh(U_o s_i + W_o E_d \hat{y}_{i-1} + b_o) \) (“output projection”)
output \( \hat{y}_i = \arg \max V_o t_i \)
Implementation: Training vs. Runtime

training: \( y_j \) (ground truth) \times \text{runtime: } \hat{y}_j \) (decoded)

\[ \begin{align*}
\text{training: } y_j & \quad \text{(ground truth)} \\
\text{runtime: } \hat{y}_j & \quad \text{(decoded)}
\end{align*} \]
Summary So Far

- Statistical MT chooses the most probable sentence:
  \[ \hat{e}_{1} = \arg\max_{I, e_{1}} p(e_{1}^{I}|f_{1}^{J}) \]

- Independence assumptions (LM vs. TM; phrase independence) were harmful.

- Neural MT predicts word by word; “just a clever LM”.
  \[ p(e_{1}^{I}|f_{1}^{J}) = \prod_{i=1}^{I} p(e_{i}|e_{1}, \ldots, e_{i-1}, f_{1}^{J}) \]
  - Sub-word units, word embeddings, RNN for variable-length, encoder-decoder.

- We moved from searching for best minimum translation units to representing words and phrases in continuous space and relating them to each other.
Attention is All You Need  (Vaswani et al., 2017)
Transformer Detailed Walkthroughs

Transformer Illustrated:

- http://jalammar.github.io/illustrated-transformer/
  Amazingly simple description! (I am reusing the pictures.)

Transformer paper annotated with PyTorch code:

- http://nlp.seas.harvard.edu/2018/04/03/attention.html
- PyTorch by examples:
  https://github.com/jcjohnson/pytorch-examples

Summary at Medium:

- https://medium.com/@adityathiruvengadam/
  transformer-architecture-attention-is-all-you-need-aecc
Self-Attention Motivation

- Sequences of arbitrary length \( n \) need to be processed.
- Information gets lost over too many processing steps.
- RNNs make the (time-unrolled) network as deep as \( n \).
- CNNs allow to trade kernel size \( k \) and depth for a target “receptive field”:
Self-Attention

• Goal: **Aggregate** arbitrary-length input to fixed-size vector. Allow **data-driven, trainable** aggregation.
Self-Attention


Given the sequence of inputs $x_1, \ldots, x_n$:

- Create three “views” of them: queries, keys, values.

![Diagram of Self-Attention]
Match All Queries with All Keys

Input

Embedding

Queries

Keys

Values

Score

Thinking

Machines

$x_1$

$q_1$

$k_1$

$v_1$

$q_1 \cdot k_1 = 112$

$x_2$

$q_2$

$k_2$

$v_2$

$q_1 \cdot k_2 = 96$
Normalize Scores

Input

Embedding

Values

Score

Divide by 8 (\(\sqrt{d_k}\))

Softmax

\[ q_1 \cdot k_1 = 112 \]

\[ q_1 \cdot k_2 = 96 \]

\[ q_1 \cdot k_3 = 14 \]

\[ q_1 \cdot k_4 = 12 \]

\[ q_1 \cdot k_5 = 0.88 \]

\[ q_1 \cdot k_6 = 0.12 \]
Aggregate Values Accordingly

Input

Embedding

Values

Softmax

Softmax $X$ Value

Sum

Thinking

Machines

$x_1$

$v_1$

$v_1$

$0.88$

$x_2$

$v_2$

$v_2$

$0.12$

$z_1$

$z_2$
Transformer = 6 Layers Enc + 6 Dec

INPUT: Je suis étudiant

OUTPUT: I am a student
Composition of One Layer

ENCODER

Feed Forward

Self-Attention

DECODER

Feed Forward

Encoder-Decoder Attention

Self-Attention
Self-Attention in Transformer

Three uses of multi-head attention in Transformer

• **Encoder-Decoder Attention:**
  • Q: previous decoder layers; K = V: outputs of encoder
    ⇒ Decoder positions attend to all positions of the input.

• **Encoder Self-Attention:**
  • Q = K = V: outputs of the previous layer of the encoder
    ⇒ Encoder positions attend to all positions of previous layer.

• **Decoder Self-Attention:**
  • Q = K = V: outputs of the previous decoder layer.
    • Masking used to prevent depending on future outputs.
    ⇒ Decoder attends to all its previous outputs.
Multi-Head Attention

Calculating attention separately in eight different attention heads

ATTENTION HEAD #0

ATTENTION HEAD #1

ATTENTION HEAD #7

Z₀

Z₁

Z₇
Self-Attention at Enc Layer #5: 1 Head
Self-Attention at Enc Layer #5: 2 Heads
Self-Attention at Enc Layer #5: 8 Heads
Some of Training Magic
Kocmi and Bojar (2017) explore curriculum learning:
- Start with simpler sentences first, add complex ones later.

![Graph showing BLEU scores vs. steps (in millions of examples)]
Curriculum vs. Catastrophic Forgetting

- Kocmi and Bojar (2017) explore curriculum learning:
  - Start with simpler sentences first, add complex ones later.
  - When “simpler” means “shorter”:
Kocmi and Bojar (2017) explore curriculum learning:
- Start with simpler sentences first, add complex ones later.
- When “simpler” means “shorter”:
  - Clear jumps in score as bins of longer sentences are allowed.
Kocmi and Bojar (2017) explore curriculum learning:

- Start with simpler sentences first, add complex ones later.
- When “simpler” means “shorter”:
  - Clear jumps in score as bins of longer sentences are allowed.
  - Reversed curriculum unlearns to produce long sentences.

![Graph showing BLEU scores over steps (in millions of examples)](attachment:graph.png)
### WMT 2018 English-to-Czech news translation results:

<table>
<thead>
<tr>
<th>System</th>
<th>Ave. %</th>
<th>Ave. z</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUNI-Transformer</td>
<td>84.4</td>
<td>0.667</td>
</tr>
<tr>
<td>UEDIN</td>
<td>79.8</td>
<td>0.521</td>
</tr>
<tr>
<td>Professional Translation</td>
<td>78.6</td>
<td>0.483</td>
</tr>
<tr>
<td>ONLINE-B</td>
<td>68.1</td>
<td>0.128</td>
</tr>
<tr>
<td>ONLINE-A</td>
<td>59.4</td>
<td>—0.178</td>
</tr>
<tr>
<td>ONLINE-G</td>
<td>54.1</td>
<td>—0.354</td>
</tr>
</tbody>
</table>
Machine Translation Surpassing Humans (1/2)

- WMT 2018 English-to-Czech news translation results:

<table>
<thead>
<tr>
<th>Ave. %</th>
<th>Ave. z</th>
<th>System</th>
</tr>
</thead>
<tbody>
<tr>
<td>84.4</td>
<td>0.667</td>
<td><strong>CUNI-Transformer</strong></td>
</tr>
<tr>
<td>79.8</td>
<td>0.521</td>
<td><strong>UEDIN</strong></td>
</tr>
<tr>
<td>78.6</td>
<td>0.483</td>
<td><strong>Professional Translation</strong></td>
</tr>
<tr>
<td>68.1</td>
<td>0.128</td>
<td><strong>ONLINE-B</strong></td>
</tr>
<tr>
<td>59.4</td>
<td>−0.178</td>
<td><strong>ONLINE-A</strong></td>
</tr>
<tr>
<td>54.1</td>
<td>−0.354</td>
<td><strong>ONLINE-G</strong></td>
</tr>
</tbody>
</table>

Caveats:
- Humans translated whole documents, MT individual segments.
  - Evaluation was done for *individual segments.*
ARTICLE

Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals

Martin Popel 1,5✉, Marketa Tomkova 2,5, Jakub Tomek 3,5, Łukasz Kaiser 4, Jakob Uszkoreit 4, Ondřej Bojar 1 & Zdeněk Žabokrtský 1

https://doi.org/10.1038/s41467-020-18073-9
Manual evaluation by domain experts, scoring in categories:

1. Language Resources – Spelling and Morphology
2. Vocabulary – Adequacy of Terms Used
3. Vocabulary – Clarity of the Text in Terms of Used Words
4. Syntax and Word Order
5. Coherence and Overall Understanding of the Text

Plotted as average rank for better comparibility.
Supplement No. 1 to the agreement on the sublease the apartment, of 13th May 2016
On the day, month and year written below Marta Burešová, pers. no. 695604/3017
Address: Radimova 8, Prague 6, 169 00 as the tenant on the one hand (Hereinafter
referred to as "the tenant") and Karolína Černá, pers. no. 136205/891 Address:
Alfrédova 13, Praha 4, 142 00 As a lessee on the other (Hereinafter referred to as ”the
lessee”) collectively also referred to as ”the Contracting parties” have agreed on this
Supplement No. 1 to the Agreement on the sublease the apartment, of 13th May 2016
(hereinafter referred to as the "Supplement No. 1")
I. Introductory Provisions
On 13th May 2016, the tenant and the lessee closed the Agreement on the sublease of
the apartment, under which the tenant let the lessee use the apartment No. 4 (area 49
m²) of size 1+1/L in the ground floor of the house in Prague 4, Alfrédova 13,
Dodatek č. 1 ke smlouvě o podnájmu bytu ze dne 13. května 2016  
V den, měsíc a rok níže napsané Marta Burešová, pers. no. 695604/3017 Adresa: Radimova 8, Praha 6, 169 00 jako nájemce na jedné straně (dále jen “nájemce”) a Karolína Černá, pers. no. 136205/891 Adresa: Alfrédova 13, Praha 4, 142 00 jako nájemce na straně druhé (dále jen “nájemce”) společně označované také jako “smluvní strany” se dohodly na tomto dodatku č. 1 ke smlouvě o podnájmu, dále jen “nájemní smlouva”, dále jen “13. května 2016”).  
I. Úvodní ustanovení  
Dne 13. května 2016 nájemce a nájemce uzavřeli smlouvu o dalším pronájmu bytu, podle níž nájemce pronajímá nájemci byt č. 4 (plocha 49 m²) o velikosti 1+1/l v přízemí domu v Praze 4, Alfrédova 13, ...
Dodatek č. 1 ke smlouvě o podnájmu bytu ze dne 13. května 2016
V den, měsíc a rok níže napsané Marta Burešová, pers. no.
695604/3017 Adresa: Radimova 8, Praha 6, 169 00 jako nájemce na jedné straně (dále jen “nájemce”) a Karolína Černá, pers. no.
136205/891 Adresa: Alfrédova 13, Praha 4, 142 00 jako nájemce na straně druhé (dále jen “nájemce”) společně označované také jako “smluvní strany” se dohodly na tomto dodatku č. 1 ke smlouvě o podnájmu, dále jen “nájemní smlouva”, dále jen “13. května 2016”).
I. Úvodní ustanovení
Dne 13. května 2016 nájemce a nájemce uzavřeli smlouvu o dalším pronájmu bytu, podle níž nájemce pronajímá nájemci byt č. 4 (plocha 49 m²) o velikosti 1+1/l v přízemí domu v Praze 4, Alfrédova 13, ...
Caveat on Interpreting Results
“Modeling Source Syntax Helps NMT” (1/2)


Forcing one Trafo head to provide dependency tree helps BLEU.

<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>36.66</td>
<td>–</td>
</tr>
<tr>
<td>Parse from layer 0</td>
<td>36.60</td>
<td>82.85</td>
</tr>
<tr>
<td>Parse from layer 1</td>
<td><strong>38.01</strong></td>
<td>90.78</td>
</tr>
<tr>
<td>Parse from layer 2</td>
<td>37.87</td>
<td>91.18</td>
</tr>
<tr>
<td>Parse from layer 3</td>
<td>37.67</td>
<td>91.43</td>
</tr>
<tr>
<td>Parse from layer 4</td>
<td>37.60</td>
<td><strong>91.56</strong></td>
</tr>
<tr>
<td>Parse from layer 5</td>
<td>37.67</td>
<td>91.46</td>
</tr>
</tbody>
</table>

• Forcing one Trafo head to provide dependency tree helps BLEU.
**“Modeling Source Syntax Helps NMT” (1/2)**

- Forcing one Trafo head to provide dependency tree helps BLEU.
- Forcing one Trafo head to provide linear tree helps more.


<table>
<thead>
<tr>
<th></th>
<th>BLEU</th>
<th>UAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>36.66</td>
<td>–</td>
</tr>
<tr>
<td>Parse from layer 0</td>
<td>38.14</td>
<td>99.96</td>
</tr>
<tr>
<td>Parse from layer 1</td>
<td>38.06</td>
<td><strong>99.99</strong></td>
</tr>
<tr>
<td>Parse from layer 2</td>
<td>37.85</td>
<td>99.98</td>
</tr>
<tr>
<td>Parse from layer 3</td>
<td>37.70</td>
<td>99.98</td>
</tr>
<tr>
<td>Parse from layer 4</td>
<td>37.47</td>
<td>99.96</td>
</tr>
<tr>
<td>Parse from layer 5</td>
<td>37.54</td>
<td>99.95</td>
</tr>
</tbody>
</table>

- Forcing one Trafo head to provide dependency tree helps BLEU.
- Forcing one Trafo head to provide linear tree helps more.

Alternating output words and CCG tags helps. (Nadejde et al. 2017)

\[
\text{Tgt: } \text{NP Obama \((S[dcl]\backslash NP)/PP)/NP receives NP Net+ an+ yahu PP/NP in NP/N the N capital (NP/N)}
\]

We tried the same with: (RNN or Transformer; interleaved or multi-decoder)
- correct CCG tags, • random tags, ⋆ a single dummy tag.

Except ⋆ single dummy tag, both improve over the • baseline.

Transfer Learning (TL)

Parent corpus

CS
EN

Child corpus

ET
EN
Transfer Learning (TL)

Parent corpus

CS EN

Child corpus

ET EN

Balanced vocabulary

CS EN + ET EN
Transfer Learning (TL)

Parent corpus

CS
EN

Child corpus

ET
EN

Balanced vocabulary

the_
že_
ying_
staying_
pra
pracovat_
...
## TL Exploits Language Similarity

### Child model: Slovak

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>9x</td>
<td>from English</td>
<td>16.13</td>
<td>17.75</td>
<td>1.62 *</td>
</tr>
<tr>
<td>Czech</td>
<td>9x</td>
<td>to English</td>
<td>19.19</td>
<td>22.42</td>
<td>3.23 *</td>
</tr>
</tbody>
</table>

* statistically significant
“TL Exploits Language Similarity”

Child model: Slovak

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>9x</td>
<td>from English</td>
<td>16.13</td>
<td>17.75</td>
<td>1.62 *</td>
</tr>
<tr>
<td>Czech</td>
<td>9x</td>
<td>to English</td>
<td>19.19</td>
<td>22.42</td>
<td>3.23 *</td>
</tr>
</tbody>
</table>

* statistically significant
## “TL Exploits Language Similarity”

### Child model: Slovak

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>9x</td>
<td>from English</td>
<td>16.13</td>
<td>17.75</td>
<td>1.62 *</td>
</tr>
<tr>
<td>Czech</td>
<td>9x</td>
<td>to English</td>
<td>19.19</td>
<td>22.42</td>
<td>3.23 *</td>
</tr>
</tbody>
</table>

* statistically significant

### Child model: Estonian

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Finnish</td>
<td>3.5x</td>
<td>from English</td>
<td>17.03</td>
<td>19.74</td>
<td>2.71 *</td>
</tr>
<tr>
<td>Russian</td>
<td>16x</td>
<td>from English</td>
<td>17.03</td>
<td>20.09</td>
<td>3.06 *</td>
</tr>
<tr>
<td>Czech</td>
<td>50x</td>
<td>from English</td>
<td>17.03</td>
<td>20.41</td>
<td>3.38 *</td>
</tr>
<tr>
<td>Finnish</td>
<td>3.5x</td>
<td>to English</td>
<td>21.74</td>
<td>24.18</td>
<td>2.44 *</td>
</tr>
<tr>
<td>Russian</td>
<td>16x</td>
<td>to English</td>
<td>21.74</td>
<td>23.54</td>
<td>1.80 *</td>
</tr>
</tbody>
</table>

* statistically significant
### TL Exploits Language Similarity

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Czech</td>
<td>9x</td>
<td>from English</td>
<td>16.13</td>
<td>17.75</td>
<td>1.62 *</td>
</tr>
<tr>
<td>Czech</td>
<td>9x</td>
<td>to English</td>
<td>19.19</td>
<td>22.42</td>
<td>3.23 *</td>
</tr>
<tr>
<td>Finnish</td>
<td>3.5x</td>
<td>from English</td>
<td>17.03</td>
<td>19.74</td>
<td>2.71 *</td>
</tr>
<tr>
<td>Russian</td>
<td>16x</td>
<td>from English</td>
<td>17.03</td>
<td>20.09</td>
<td>3.06 *</td>
</tr>
<tr>
<td>Czech</td>
<td>50x</td>
<td>from English</td>
<td>17.03</td>
<td>20.41</td>
<td>3.38 *</td>
</tr>
<tr>
<td>Finnish</td>
<td>3.5x</td>
<td>to English</td>
<td>21.74</td>
<td>24.18</td>
<td>2.44 *</td>
</tr>
<tr>
<td>Russian</td>
<td>16x</td>
<td>to English</td>
<td>21.74</td>
<td>23.54</td>
<td>1.80 *</td>
</tr>
</tbody>
</table>

* statistically significant
## TL Exploits Language Similarity

### Child model: Slovak

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related</td>
<td>9x</td>
<td>from English</td>
<td>16.13</td>
<td>17.75</td>
<td>1.62 *</td>
</tr>
<tr>
<td>Related</td>
<td>9x</td>
<td>to English</td>
<td>19.19</td>
<td>22.42</td>
<td>3.23 *</td>
</tr>
</tbody>
</table>

### Child model: Estonian

<table>
<thead>
<tr>
<th>Parent model</th>
<th>Corpus size difference</th>
<th>Direction</th>
<th>Baseline (BLEU)</th>
<th>Transfer (BLEU)</th>
<th>Δ (BLEU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Related</td>
<td>3.5x</td>
<td>from English</td>
<td>17.03</td>
<td>19.74</td>
<td>2.71 *</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>16x</td>
<td>from English</td>
<td>17.03</td>
<td>20.09</td>
<td>3.06 *</td>
</tr>
<tr>
<td>Biggest</td>
<td>50x</td>
<td>from English</td>
<td>17.03</td>
<td>20.41</td>
<td>3.38 *</td>
</tr>
<tr>
<td>Related</td>
<td>3.5x</td>
<td>to English</td>
<td>21.74</td>
<td>24.18</td>
<td>2.44 *</td>
</tr>
<tr>
<td>Cyrillic</td>
<td>16x</td>
<td>to English</td>
<td>21.74</td>
<td>23.54</td>
<td>1.80 *</td>
</tr>
</tbody>
</table>

* statistically significant
Summary

- Given all the data we had for SMT, given big GPUs, given all the training tricks, and given a few weeks of training time, Transformer can reach (average) quality comparable to (sloppy) humans.

- Intuition why something works is often wrong.
  - Use trivial baselines to exclude misinterpretations; read more here:

References


Ondřej Bojar, Miroslav Janíček, Zdeněk Žabokrtský, Pavel Češka, and Peter Beňa. 2008. CzEng 0.7: Parallel Corpus with Community-Supplied Translations. In *Proceedings of the Sixth International Language Resources and Evaluation (LREC’08)*, Marrakech, Morocco, May. ELRA.


