Neural Networks & Translation

Andy Way*
ADAPT Centre, Dublin City University

andy.way@adaptcentre.ie

*With much gratitude to Marcello Federico, John Kelleher & Philipp Koehn for some (excellent!) slides
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
What is a function?

A function maps a set of inputs (numbers) to an output (number)

\[ \text{sum}(2, 5, 4) \rightarrow 11 \]
What is a \texttt{WEIGHTEDSUM} function?

\[
\text{WEIGHTEDSUM}([n_1, n_2, \ldots, n_m], [w_1, w_2, \ldots, w_m])
\]

Input Numbers \hspace{3cm} Weights

\[
= (n_1 \times w_1) + (n_2 \times w_2) + \cdots + (n_m \times w_m)
\]

\[
\text{WEIGHTEDSUM}([3, 9], [\ -3, 1])
\]

\[
= (3 \times -3) + (9 \times 1)
\]

\[
= -9 + 9
\]

\[
= 0
\]
What is an **ACTIVATION** function?

An **ACTIVATION** function takes the output of our **WEIGHTEDSUM** function and applies another mapping to it.
What is an \textbf{ACTIVATION} function?

\[
\text{ACTIVATION} = \text{LOGISTIC}(\text{WEIGHTEDSUM}(\{n_1, n_2, \ldots, n_m\}, [w_1, w_2, \ldots, w_m]))
\]

\[
\begin{align*}
\text{LOGISTIC}(\text{WEIGHTEDSUM}([3, 9], [-3, 1])) &= \text{LOGISTIC}((3 \times -3) + (9 \times 1)) \\
&= \text{LOGISTIC}(-9 + 9) \\
&= \text{LOGISTIC}(0) \\
&= 0.5
\end{align*}
\]
What is a Neuron?

The simple list of operations that we have just described defines the fundamental building block of a neural network: the Neuron.

Neuron =
activation(weightedSum([n_1, n_2, \ldots, n_m], [w_1, w_2, \ldots, w_m]))

Input Numbers  Weights
Inspired by Biological Neurons

Impulses carried toward cell body

Dendrite
Cell body
Axon
Myelin sheath
Node of Ranvier
Synapse/Axon Terminal
Schwann cell

Impulses carried away from cell body

Image credit: http://cs231n.github.io/neuralnetworks-1/
Overview

• Basic building blocks: neurons
• **Neural Networks**
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• **Neural Machine Translation: Architecture**
  – Encoders
  – Decoders (language models)
  – Attention
• **Neural Machine Translation: Improvements**
• Future Work in NMT
• Concluding Remarks
What is a Neural Network?
Where do the \textbf{WEIGHTS} come from?
So we need:
So we need:

A vector of input nodes
So we need:

- A vector of input nodes
- A vector of hidden nodes
So we need:

- A vector of input nodes
- A vector of hidden nodes
- A vector of output nodes
So we need:

- A vector of input nodes
- A vector of hidden nodes
- A vector of output nodes
- A matrix of weights connecting input nodes and hidden nodes
So we need:

- A vector of input nodes
- A vector of hidden nodes
- A vector of output nodes
- A matrix of weights connecting input nodes and hidden nodes
- A matrix of weights connecting hidden nodes and output nodes
Training a **Neural Network**

- We train a neural network by iteratively updating the weights.
- We start by randomly assigning weights to each edge.
- We then show the network examples of inputs and expected outputs and update the weights using **Backpropagation** so that the network outputs match the expected outputs.
- We keep updating the weights until the network is working the way we want.
Feed-forward neural networks

Machine Learning

Input

Output

Input layer

Hidden layer

Hidden layer

Output layer
Feed-forward neural networks

NB, this is a deep neural network
Feed-forward neural networks

Is this really me?
Feed-forward neural networks
Feed-forward neural networks
Feed-forward neural networks
Feed-forward neural networks

NB, something new, **bias nodes**. Always have value 1; give the network something to work with in case all input values are 0 - avoids weighted sum being equal to 0
Feed-forward neural networks

\[ \sigma \text{ (input x weights)} = \sigma(2.7) = 0.94 \]
Other common activation functions

\[ \tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \]

\[ \text{relu}(x) = \max(0, x) \]

\[ \text{softmax}(x)_i = \frac{e^{x_i}}{\sum e^{x_i}} \]

Useful for modeling probability (in classification task)
Feed-forward neural networks
Feed-forward neural networks
Feed-forward neural networks
Feed-forward neural networks

Class “CAT” wins because it got the largest value.
FF-NNs: multilayer perceptron

Weight matrices and bias vectors are the network parameters (bias nodes in the hidden layers are not shown in the picture)
How do NNs learn a task?

Once the architecture of the network is fixed its behaviour depends only on the values of its weights and biases.

We start with random W&Bs and progressively adjust them with trial and error over a labelled data set.
# Labeled data set

<table>
<thead>
<tr>
<th>Training</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Cat" /> <img src="image2" alt="Cat" /> <img src="image3" alt="Cat" /> <img src="image4" alt="Cat" /> <img src="image5" alt="Cat" /></td>
<td><img src="image6" alt="Cat" /> <img src="image7" alt="Cat" /></td>
</tr>
<tr>
<td><img src="image8" alt="Owl" /> <img src="image9" alt="Owl" /> <img src="image10" alt="Owl" /> <img src="image11" alt="Owl" /> <img src="image12" alt="Owl" /></td>
<td><img src="image13" alt="Owl" /> <img src="image14" alt="Owl" /></td>
</tr>
<tr>
<td><img src="image15" alt="Dog" /> <img src="image16" alt="Dog" /> <img src="image17" alt="Dog" /> <img src="image18" alt="Dog" /> <img src="image19" alt="Dog" /></td>
<td><img src="image20" alt="Dog" /> <img src="image21" alt="Dog" /></td>
</tr>
<tr>
<td><img src="image22" alt="Lion" /> <img src="image23" alt="Lion" /> <img src="image24" alt="Lion" /> <img src="image25" alt="Lion" /> <img src="image26" alt="Lion" /></td>
<td><img src="image27" alt="Lion" /> <img src="image28" alt="Lion" /></td>
</tr>
</tbody>
</table>
Learning by trial and error

Label

Output

\[ W^1 \]

\[ W^2 \]

\[ b^2 \]

\[ W^3 \]

\[ b^3 \]

Input

Updated W&B
Learning by trial and error

Label
Output

Error Function

Update Function

\( W^0 \)
\( W^1 \)
\( b^0 \)
\( b^1 \)

Input

New W&B

\( W^0 \)
\( W^1 \)
\( b^0 \)
\( b^1 \)
Learning by trial and error

Input

Label

Output

\[ W^3 \]
\[ W^2 \]
\[ b^2 \]
\[ W^1 \]
\[ b^1 \]

Error Function

Update Function

\[ W^3 \]
\[ W^2 \]
\[ b^2 \]

New W&B

38 of 119
Learning by trial and error

After many, many,...,many training sessions ...
Learning by trial and error

### Diagram

- **Label**
- **Output**
- **Input**
- **Error Function**
- **No Update!**

### Mathematical Symbols

- $W^L$
- $W^2$
- $b^2$
- $W^1$
- $b^1$

### Text

The diagram illustrates a neural network structure with layers labeled $W^L$, $W^2$, and $W^1$ and biases $b^2$ and $b^1$. The process of learning by trial and error is depicted through the interaction between the label, output, error function, and the decision to update the weights and biases.
What about the test set?

We got 75% accuracy!
How do we represent words?

One-hot vectors:

- dog = (0,0,0,0,1,0,0,0,0,0,...) T
- cat = (0,0,0,0,0,0,0,0,1,0,...) T
- eat = (0,1,0,0,0,0,0,0,0,0,...) T
- etc ...

• multidimensional space, huge vectors, so limit vocabulary to (say) 20K tokens, OTHER for rest
Similar words behave similarly!

• but the cute dog jumped
• but the cute cat jumped

• child hugged the cat tightly
• child hugged the dog tightly

• like to watch cat videos
• like to watch dog videos

cf. Firth’s “you shall know a word by the company it keeps”
Word Embeddings

Each word is represented by a vector of numbers that positions the word in a multi-dimensional space, e.g.:

\[ \text{king} = \langle 55, -10, 176, 27 \rangle \]
\[ \text{man} = \langle 10, 79, 150, 83 \rangle \]
\[ \text{woman} = \langle 15, 74, 159, 106 \rangle \]
\[ \text{queen} = \langle 60, -15, 185, 50 \rangle \]
Word Embeddings

\[
\text{vec}(\text{King}) - \text{vec}(\text{Man}) + \text{vec}(\text{Woman}) \approx \text{vec}(\text{Queen})^2
\]

\(^2\) Linguistic Regularities in Continuous Space Word Representations (Mikolov et al., 2013)
Word Embedding using LM Prediction

Figure 13.11: Full architecture of a feed-forward neural network language model. Context words \( (w_{t-4}, w_{t-3}, w_{t-2}, w_{t-1}) \) are represented in a one-hot vector, then projected into continuous space as word embeddings (using the same weight matrix \( C \) for all words). The predicted word is computed as a one-hot vector via a hidden layer.
Word Embedding using LM Prediction

- Enable generalization between words (*clustering*)
- Robust prediction in unseen context (*back-off*)
Are feed-forward NLMs restricted?

- FFLMs can use (much) longer contexts than traditional statistical back-off models. e.g. up to 30-gram models are being used!
Are feed-forward NLMs restricted?

• FFLMs can use (much) longer contexts than traditional statistical back-off models. e.g. up to 30-gram models are being used!
• But we are still using a fixed content word window (5-gram in the previous example) ...
Are feed-forward NLMs restricted?

• FFLMs can use (much) longer contexts than traditional statistical back-off models. e.g. up to 30-gram models are being used!
• But we are still using a fixed content word window (5-gram in the previous example) …
• Recurrent NLMs can condition on any length context sequences by reusing the hidden layer when predicting \( w_n \) as additional input to predict word \( w_{n-1} \)
Recurrent Neural Networks

- RNNs useful for processing sequential input (like language)
- Using an RNN we process our sequential data one input at a time
- In an RNN the outputs of some of the neurons for one input is fed back into the network as part of the next input
- The ‘copy value’ encodes the previous context — part of a sentence in MT — in the sequence

Figure 13.13: Recurrent neural language models: After predicting Word 2 in the context of following Word 1, we re-use this hidden layer (alongside the correct Word 2) to predict Word 3. Again, the hidden layer of this prediction is re-used for the prediction of Word 4.
Recurrent Neural Networks

Diagram:
- Output
- Hidden
- Memory
- Input
- $y_t$
- $h_t$
- $h_{t-1}$
- $x_t$
Recurrent neural networks

this layer is connected with itself!
Time Unfolded RNN

the cat on the mat
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
Machine Translation

Source → Machine Learning → Target

English

German
Phrase-based vs. Neural MT

Morgen fliege ich nach Italien zur Konferenz

Tomorrow I will fly to the conference in Italy
NMT: encoder-decoder

the cat on the mat
NMT: encoder-decoder

deckard

cat

don

the

mat
NMT: encoder-decoder

encoder final state represents the whole sentence!
NMT: encoder-decoder

the cat on the mat

<s>
NMT: encoder-decoder

Softmax operation maps scores to probabilities
Word Predictions in NMT

Input Sentence
ich glaube aber auch, er ist clever genug um seine Aussagen tage genug zu halten, so dass sie auf verschiedene Art und Weise interpretiert werden können.

Output Word Predictions

<table>
<thead>
<tr>
<th>Best</th>
<th>Alternatives</th>
</tr>
</thead>
<tbody>
<tr>
<td>but</td>
<td>however (25.3%), I (20.4%), yet (1.9%), and (0.8%), nor (0.8%), ...</td>
</tr>
<tr>
<td>I</td>
<td>also (6.0%), , (4.7%), it (1.2%), in (0.7%), nor (0.5%), he (0.4%), ...</td>
</tr>
<tr>
<td>also</td>
<td>think (4.2%), do (3.1%), believe (2.9%), , (0.8%), too (0.5%), ...</td>
</tr>
<tr>
<td>believe</td>
<td>think (28.6%), feel (1.6%), do (0.8%), ...</td>
</tr>
<tr>
<td>he</td>
<td>that (6.7%), it (2.2%), him (0.2%), ...</td>
</tr>
<tr>
<td>is</td>
<td>'s (24.4%), has (0.3%), was (0.1%), ...</td>
</tr>
<tr>
<td>clever</td>
<td>smart (0.6%), ...</td>
</tr>
<tr>
<td>enough</td>
<td>(99.9%)</td>
</tr>
<tr>
<td>to</td>
<td>about (1.2%), for (1.1%), in (1.0%), of (0.3%), around (0.1%), ...</td>
</tr>
<tr>
<td>keep</td>
<td>maintain (4.5%), hold (4.4%), be (4.2%), have (1.1%), make (1.0%), ...</td>
</tr>
<tr>
<td>his</td>
<td>its (2.1%), statements (1.5%), what (1.0%), out (0.6%), the (0.6%), ...</td>
</tr>
<tr>
<td>statements</td>
<td>testimony (1.5%), messages (0.7%), comments (0.6%), ...</td>
</tr>
<tr>
<td>vague</td>
<td>(91.9%)</td>
</tr>
<tr>
<td>enough</td>
<td>(96.2%), vz@ (1.2%), in (0.6%), ambiguous (0.3%), ...</td>
</tr>
<tr>
<td>so</td>
<td>and (0.2%), ...</td>
</tr>
<tr>
<td>they</td>
<td>(51.1%), , (44.3%), to (1.2%), in (0.6%), and (0.5%), just (0.2%), that (0.2%), ...</td>
</tr>
<tr>
<td>can</td>
<td>that (35.3%), it (2.5%), can (1.6%), you (0.8%), we (0.4%), to (0.3%), ...</td>
</tr>
<tr>
<td>be</td>
<td>may (2.7%), could (1.6%), are (0.8%), will (0.6%), might (0.5%), ...</td>
</tr>
<tr>
<td>be</td>
<td>have (0.3%), interpret (0.2%), get (0.2%), ...</td>
</tr>
<tr>
<td>interpreted</td>
<td>interpre@ (0.1%), constru@ (0.1%), ...</td>
</tr>
<tr>
<td>in</td>
<td>(96.5%), on (0.9%), differently (0.5%), as (0.3%), to (0.2%), for (0.2%), by (0.1%), ...</td>
</tr>
<tr>
<td>different</td>
<td>a (25.2%), various (22.7%), several (3.6%), ways (2.4%), some (1.7%), ...</td>
</tr>
<tr>
<td>ways</td>
<td>way (0.2%), manner (0.2%), ...</td>
</tr>
<tr>
<td>.</td>
<td>&lt;/s&gt; (0.2%), , (0.1%), ...</td>
</tr>
</tbody>
</table>

Figure 13.25: Word predictions of the neural machine translation model. Frequently, most of the probability mass is given to the top choice, but semantically related words may rank high, e.g., believe (68.4%) vs. think (28.6%). The subword units interpre@ are explain in Section 13.6.2 on page 61.
Word embedding

![Diagram of word embedding with examples of words and their embeddings]
NMT: encoder-decoder
NMT: encoder-decoder

Entire translation is generated from the final encoder state!
NMT with attention model

Each translation is generated from specific encoder states!
NMT with attention model
NMT with attention model

Bidirectional RNN (Bahdanau et al., 2015)
A neural net for handwriting recognition
PBMT and NMT learn differently

NMT tends to be better than PBMT on new sentences.

The opposite happens on inputs very similar to training examples.

Thanks to Marcello Federico for this observation.
PBMT and NMT learn differently

NMT have a long history, longer than PBMT, but started to outperform PBMT in Fall 2015

• Which are the strengths of NMT?
• On which linguistic phenomena?
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
IWSLT 2015 MT Task
IWSLT 2015 MT Task

Evaluation Data

**tst 2015 HE SET**
12 TED Talks
- 600 src sentences
- ~10K src words
IWSLT 2015 MT Task

Evaluation Data

**tst 2015 HE SET**
12 TED Talks
- 600 src sentences
- ~10K src words

- **PBSY** (Huck & Birch, '15)
- **HPB** (Jehl et al., '15)
- **SPB** (Ha et al., '15)
- **NMT** (Luong & Manning, '15)
IWSLT 2015 MT Task

Evaluation Data

*tst 2015 HE SET*
12 TED Talks
- 600 src sentences
- ~10K src words

- PBSY (Huck & Birch, '15)
- HPB (Jehl et al., '15)
- SPB (Ha et al.,’15)
- NMT (Luong & Manning, ‘15)

Post-Edit Data

- PBSY Post-Edit
- HPB Post-Edit
- SPB Post-Edit
- NMT Post-Edit
Evaluation Metrics

**TER:** traces the edits done by post-editors

Reliable and informative since post-editors’ variability is controlled

*Analyses of overall quality*

cf. Bentivogli et al. (EMNLP, 2016)
**Evaluation Metrics**

**TER:** traces the edits done by post-editors

---

Focuses on what a human implicitly annotated as a translation error

**Analyses of errors**
Overall Quality

![Bar chart showing Overall Quality for HPB, SPB, PBSY, and NMT]
Analyses of Errors

• Lexical

• Morphology

• Word Order
Lexical Errors

![Bar chart showing HTER for HPB, SPB, PBSY, and NMT with Lemmas as the category on the y-axis and the names of the categories on the x-axis. The bars represent the number of lexical errors.](image)
Lexical Errors

- 17%
Morphology Errors

![Bar chart showing HTER values for HPB, SPB, PBSY, and NMT, with two error types: Words and Lemmas.](chart.png)
Morphology Errors

<table>
<thead>
<tr>
<th></th>
<th>Morphology Errors</th>
<th>Lemmas</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPB</td>
<td>18.4%</td>
<td></td>
</tr>
<tr>
<td>SPB</td>
<td>18%</td>
<td></td>
</tr>
<tr>
<td>PBSY</td>
<td>16.9%</td>
<td></td>
</tr>
<tr>
<td>NMT</td>
<td>13.7%</td>
<td></td>
</tr>
</tbody>
</table>

HTER
Morphology Errors

![Bar chart showing HTER percentages for different models: HPB (18.4%), SPB (18%), PBSY (16.9%), and NMT (13.7%). The chart highlights a 19% decrease.]
Reordering Errors

<table>
<thead>
<tr>
<th># SHIFTS</th>
<th>HPB</th>
<th>SPB</th>
<th>PBSY</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>#</td>
<td>3.6</td>
<td>3.5</td>
<td>3.1</td>
<td>1.5</td>
</tr>
</tbody>
</table>
Reordering Errors

-50%
Word Order Errors

- Do reordering errors concentrate on specific linguistic constructions?
- What kind of reordering patterns are best modeled by NMT?

⇒ Classify *shifts* by POS
# Word Order Errors

<table>
<thead>
<tr>
<th>SRC</th>
<th>I have never landed in Seattle on Friday night</th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td>Ich habe <em>gelandet</em> nie am Freitagabend in Seattle</td>
</tr>
<tr>
<td>PE</td>
<td>Ich bin nie am Freitagabend in Seattle <em>gelandet</em></td>
</tr>
</tbody>
</table>

*SHIFT*
Word Order Errors

<table>
<thead>
<tr>
<th>SRC</th>
<th><em>I have never landed in Bologna on Friday night</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>MT</td>
<td><em>Ich habe <em>gelandet</em> nie am Freitagabend in Bologna</em></td>
</tr>
<tr>
<td>PE</td>
<td>*Ich bin nie am Freitagabend in Bologna <em>gelandet</em></td>
</tr>
<tr>
<td></td>
<td>PRO AUX ADV PREP N   PREP N   V</td>
</tr>
<tr>
<td></td>
<td><strong>SHIFT</strong></td>
</tr>
</tbody>
</table>
Word Order Errors

• Largest gain of NMT over PBMT is observed on reordering of VERBS (-70% NMT-vs-PBSY)
**Word Order Errors**

- Largest gain of NMT over PBMT is observed on reordering of VERBS (-70% NMT-vs-PBSY)

<table>
<thead>
<tr>
<th>SRC</th>
<th><em>when [ ] owners</em> <strong>look</strong> <em>at this information streaming</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>PBSY</td>
<td><em>wenn [ ] Eigentümer</em> <strong>betrachten</strong> <em>diese Information Streaming</em> ✗</td>
</tr>
<tr>
<td>PE</td>
<td><em>wenn [ ] Eigentümer dieses Informations-Streaming</em> <strong>betrachten</strong> ...</td>
</tr>
<tr>
<td>NMT</td>
<td><em>wenn [ ] Besitzer sich diese Informationen</em> <strong>anschauen</strong> ... ✓</td>
</tr>
<tr>
<td>PE</td>
<td><em>wenn [ ] Besitzer sich diese Informationen</em> <strong>anschauen</strong> ...</td>
</tr>
</tbody>
</table>
Word Order Errors

- Largest gain of NMT over PBMT is observed on reordering of VERBS (-70% NMT-vs-PBSY)

<table>
<thead>
<tr>
<th>SRC</th>
<th>… individuals were shown hundreds of hours of [] videos</th>
</tr>
</thead>
<tbody>
<tr>
<td>HPB</td>
<td>Individuen <em>gezeigt wurden</em> Hunderte von Stunden [] Videos ✗</td>
</tr>
<tr>
<td>PE</td>
<td><em>wurden</em> Individuen Hunderte von Stunden [] Videos <em>gezeigt</em></td>
</tr>
<tr>
<td>NMT</td>
<td><em>wurden</em> Individuen hunderte Stunden [] Videos <em>gezeigt</em> ✓</td>
</tr>
<tr>
<td>PE</td>
<td><em>wurden</em> Individuen hunderte Stunden [] Videos <em>gezeigt</em></td>
</tr>
</tbody>
</table>
Word Order Errors

• Smallest gains observed on **PREPOSITIONS** (-18%), **NEG. PARTICLES** (17%) and **ARTICLES** (-4%)
Word Order Errors

- Smallest gains observed on **PREPOSITIONS** (-18%), **NEG. PARTICLES** (17%) and **ARTICLES** (-4%).

<table>
<thead>
<tr>
<th>SRC</th>
<th>so like many [], I’ve lived in a few closets <em>in my life</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>SPB</td>
<td>so wie viele [], ich habe in ein paar Schränke <em>in meinem Leben</em> gelebt ✗</td>
</tr>
<tr>
<td>PE</td>
<td><em>so habe ich wie viele [] während meines Lebens</em> in einigen Verstecken gelebt</td>
</tr>
<tr>
<td>NMT</td>
<td><em>wie viele [] habe ich in ein paar Schränke in meinem Leben</em> gelebt ✗</td>
</tr>
<tr>
<td>PE</td>
<td><em>wie viele [] habe ich in meinem Leben</em> in ein paar Schränken gelebt</td>
</tr>
</tbody>
</table>
### Word Order Errors

- Smallest gains observed on **PREPOSITIONS (-18%), NEG. PARTICLES (17%)** and **ARTICLES (-4%)**

<table>
<thead>
<tr>
<th>SRC</th>
<th><em>that that just did not work for systematic reasons</em></th>
</tr>
</thead>
<tbody>
<tr>
<td>HPB</td>
<td>dass nur aus systematischen Gründen <strong>nicht funktionieren</strong> ✓</td>
</tr>
<tr>
<td>PE</td>
<td><em>dass es einfach aus systematischen Gründen nicht funktioniert</em></td>
</tr>
<tr>
<td>NMT</td>
<td>dass das einfach <strong>nicht</strong> aus systematischen Gründen <strong>funktionierte</strong> ✗</td>
</tr>
<tr>
<td>PE</td>
<td>dass das aus systematischen Gründen <strong>nicht funktioniert</strong></td>
</tr>
</tbody>
</table>
Evaluation Summary

• Generic text features:
  😞 Input length: degrades faster with long sentences
  🌟 Performs better with lexically rich text

• Linguistic errors:
  😞 Lexical: -17%, Morphology: -19%, Word Order: -50%

• Word order error types
  😞 Verb reordering -70%
  More subtle translation decisions (order of semantic arguments, focus of negation) remain a challenge.
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
Hard to Predict!

Things that did not exist on Thanksgiving 10 years ago:

Uber
Airbnb
Instagram
Snapchat
Bitcoin
iPad
Kickstarter
Pinterest
App Store
Angry Birds
Slack
Siri
Lyft
Google Chrome
WhatsApp
Venmo
Candy Crush
Alexa
Tinder
Stripe
Square
Apple Watch
FB Messenger
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...
Hard to Predict!

• NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...  
• Despite claims to the contrary, problems of multilinguality aren’t going away ...
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...
- Despite claims to the contrary, problems of multilinguality aren’t going away ...
- Human Factors topics will always be current and topical
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...
- Despite claims to the contrary, problems of multilinguality aren’t going away ...
- Human Factors topics will always be current and topical
- Evaluation remains an unsolved topic
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...
- Despite claims to the contrary, problems of multilinguality aren’t going away ...
- Human Factors topics will always be current and topical
- Evaluation remains an unsolved topic
- Data Selection is arguably a more important topic in NMT than it's been heretofore in SMT
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time ...
- Despite claims to the contrary, problems of multilinguality aren’t going away ...
- Human Factors topics will always be current and topical
- Evaluation remains an unsolved topic
- Data Selection is arguably a more important topic in NMT than it's been heretofore in SMT
- User-Generated Content will continue to increase
Hard to Predict!

- NMT wasn't *at all* mainstream even three years ago, so it's very hard to predict where we'll be in (say) 3-10 years time …
- Despite claims to the contrary, problems of multilinguality aren’t going away …
- Human Factors topics will always be current and topical
- Evaluation remains an unsolved topic
- Data Selection is arguably a more important topic in NMT than it's been heretofore in SMT
- User-Generated Content will continue to increase
- Engine training times need to come down, so parallelization might be an important topic
Symbiotic Translation

MT seamlessly
• adapts to user data
• learns from post-editing

User enjoys
• enhanced productivity
• better user experience
Adaptive Neural MT
Neural Automatic Post-Editing

Police have clashed with opposition supporters, some of whom have been blocking access to polling stations.

La polizia si sono scontrati con i sostenitori dell'opposizione, alcuni dei quali hanno bloccato l'accesso ai seggi elettorali.

La polizia si è scontrata con i sostenitori dell'opposizione, alcuni dei quali hanno bloccato l'accesso ai seggi elettorali.
Multilingual MT

Diagram showing the flow of languages through an M-NMT Engine:
- Italian
- English
- Romanian
- Italian
- English
- Romanian
Overview

• Basic building blocks: neurons
• Neural Networks
  – Feed-Forward Neural Networks
  – Word Embeddings
  – Recurrent Neural Networks
• Neural Machine Translation: Architecture
  – Encoders
  – Decoders (language models)
  – Attention
• Neural Machine Translation: Improvements
• Future Work in NMT
• Concluding Remarks
Concluding Remarks

- Neural MT has rapidly kick-started a new era
  - Significant improvement in performance
  - But exaggerated and hysterical claims

- We are living in *turbulent times*:
  - Novel ideas coming out almost every week!
  - DL is offering new powerful tools
  - It will take time to maximise their full potential

- MT remains a very hard problem to tackle!
Other Applications

Images & Video
- Flickr
- Google
- YouTube

Text & Language
- Wikipedia
- Reuters
- Associated Press

Speech & Audio
- Gene Expression

Relational Data/Social Network
- Amazon
- Netflix
- eBay
- Facebook
- Twitter

Climate Change

Geological Data
MT vs. Self-Driving Cars

Technical translation
MT vs. Self-Driving Cars

Technical translation

Creative translation
MT vs. Self-Driving Cars

Technical translation

Creative translation

Difficult translation
Thanks for listening!
References


Bentivogli et al., 2016] Neural versus. Phrase-Based Machine Translation Quality: A Study. EMNLP.

Farajian et al., 2016] Neural vs. Phrase-Based Machine Translation in a Domain Scenario. EACL.

Farajian et al., 2017] Multi-Domain Neural Machine Translation through Unsupervised Adaptation. WMT.

Gehring et al., 2016] Neural Machine Translation of Rare Words with Subword Units, ACL.

Ataman et al. 2017] Linguistically Motivated Vocabulary Reduction for Neural Machine Translation from Turkish to English, EAMT.


Sato et al., 2016] Deep Architectures for Neural Machine Translation, WMT.