Neural MT Basics I

David Vilar
dvilar@amazon.com

MT Marathon 2018
3. September 2018
Welcome to the MT Marathon!
Goal: Introduce the encoder-decoder architecture.

Roadmap: What we will see in this lecture:

- Neural language models.
- Word embeddings.
- Recurrent neural networks (including LSTMs).
- Encoder-Decoder architecture.
- Comparison with previous approaches.
Goal: Introduce the encoder-decoder architecture.

Roadmap: What we will see in this lecture:
- Neural language models.
- Word embeddings.
- Recurrent neural networks (including LSTMs).
- Encoder-Decoder architecture.
- Comparison with previous approaches.

Follow-up: What you will see in the next lecture:
- Attention.
- Advanced models.
Introduction
A brief (and simplified) timeline:

1949  Shannon/Weaver: statistical approach.


1969  Chomsky: Ban on statistics.

The notion “probability of a sentence” is an entirely useless one, under any known interpretation of this term.


2014-??  Hype of deep learning approaches.
Progress in MT

From ACL tutorial by Luong, Cho and Manning 2016
Preliminaries
Roadmap

- Neural networks.
- Language models.
Linear transformation followed by a non linearity:

\[ y = f \left( \sum_k w_k x_k + b \right) \]
Linear transformation followed by a non linearity:

\[ y = f \left( \sum_k w_k x_k + b \right) \]

For a whole layer:  \( y = f (Wx + b) \)
Non-linear Functions

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

\[ \text{tanh}(x) \]

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

\[ \text{htanh}(x) \]
Multi-layer FF Networks

- Several layers, the output of one layer is the input of the next one:
  \[ y^{(l)}(x) = f(W^{(l)}y^{(l-1)}(x) + b^{(l)}) \]

- Output layer usually is a softmax operation:
  \[ p(Y = i|x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \]

- Training: error backpropagation.
  - “Smart use of chain rule”
A language model is a probability distribution over sentences:

\[ p(w_1w_2 \ldots w_N) = p(w_1^N) \]
A language model is a probability distribution over sentences:

\[ p(w_1w_2 \ldots w_N) = p(w_1^N) \]

It can be decomposed according to the chain rule:

\[ p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_1^{n-1}) \]

**Note:** mathematical equality.
Until not so long ago...

- **k-th order Markov assumption**: \((k + 1)\)-grams:

\[
p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_{1}^{n-1})
\]

\[
\approx \prod_{n=1}^{N} p(w_n|w_{n-k}^{n-1})
\]

- "Big tables" of probabilities.
- **ML estimation**: relative frequencies.
  - Smoothing for unseen events.

[Kneser and Ney, 1995, Chen and Goodman, 1996]
Motivation: ASR

Example from the Wall Street Journal 5K task:

<table>
<thead>
<tr>
<th>LM</th>
<th>Recognized</th>
<th>errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-gram</td>
<td>h ih t s eh n uh t ur z n ih g oh sh ee ey t</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>ih ng -- s ey l -- s ur t un aa s eh t s aw</td>
<td></td>
</tr>
<tr>
<td></td>
<td>n t uh b r oh k ur ih j y ooh n ih t s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HIT SENATORS NEGOTIATING SALE</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>CERTAIN ASSETS ONTO BROKERAGE UNIT’S</td>
<td></td>
</tr>
</tbody>
</table>
Example from the Wall Street Journal 5K task:

<table>
<thead>
<tr>
<th>LM</th>
<th>Recognized</th>
<th>errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-gram</td>
<td>ih t s s eh n ih t ih z n ih g oh sh ee ey t ih ng -- s ey l -- s ur t un aa s eh t s aw v dh uh b r oh k ur ih j y ooh n ih t ITS SENATE IS NEGOTIATING SALE CERTAIN ASSETS OF THE BROKERAGE UNIT</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td></td>
<td>5</td>
</tr>
</tbody>
</table>
### Example from the Wall Street Journal 5K task:

<table>
<thead>
<tr>
<th>LM</th>
<th>Recognized</th>
<th>errors</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-gram</td>
<td>ih t s eh d ih t ih z n ih g oh sh ee ey t ih ng dh uh s ey l aw v s ur t un aa s eh t s aw v dh uh b r oh k ur ih j y ooh n ih t IT SAID IT IS NEGOTIATING THE SALE OF CERTAIN ASSETS OF THE BROKER-AGE UNIT</td>
<td>0</td>
</tr>
</tbody>
</table>
Word choice:

I withdrew money from the bank.

Saqué dinero del *banco*.

[As opposed to “orilla” (→ riverbank)]
Die italienische Regierung will nicht mehr allein für die Flüchtlinge auf den Schiffen der EU-Mission Sophia verantwortlich sein.

⇓

The Italian government wants not any more alone for the refugees aboard the ship of the EU mission Sophia responsible be.
Die italienische Regierung will nicht mehr allein für die Flüchtlinge auf den Schiffen der EU-Mission Sophia verantwortlich sein.

The Italian government wants not any more alone for the refugees aboard the ship of the EU mission Sophia responsible be.

The Italian government does not want to be responsible any more for the refugees aboard the ship of the EU mission Sophia.
Neural Language Models
• *n*-gram *neural* language models.
• How to represent words.
• Dropping the Markov assumption (vanilla RNNs).
• LSTMs.
Feed-forward LM

- Trigram model
  \[ p(w_n|w_{n-2}, w_{n-1}) \].
- Prediction of current word given history.
- No 0 probabilities.
1-hot encodings

• 1-hot encoding is the “natural” way to encode symbolic information (e.g. words).
• But:
  • The encoding itself is arbitrary (e.g. first appearance of a word in the training text).
  • No useful information can be read from the vector representation.
• Example:

  the green dog bites the cat

  the  (1,0,0,0,0)
  green (0,1,0,0,0)
  dog   (0,0,1,0,0)
  bites (0,0,0,1,0)
  cat   (0,0,0,0,1)
What happens in the first layer of the network?

- Usually simplified form

\[ y^{(1)}(x) = W^{(1)}x \]

where \( x \) is a 1-hot vector.

- Multiplication reduces to column lookup.

- Maps words into continuous vectors.
Excursion: Word Embeddings

How do these word vectors look like?

- **Word embedding**: mapping of words (discrete) into a continuous space.
- Arises naturally when dealing with 1-hot encodings.
- Can be trained separately.
  - Active area of research.
  - Big improvements on some tasks.
Excursion: The most “stupid” network

\[ \text{x} \]

\[ \text{x} \]
Excursion: The most “stupid” network
Excursion: The most “stupid” network

If the “stupid” network has no errors:

- We mapped an 12-dimensional (sparse?) vector into a 4-dimensional dense vector.
Excursion: The most “stupid” network

If the “stupid” network has no errors:

• We mapped an 12-dimensional (sparse?) vector into a 4-dimensional dense vector.

However:

• The representation is still arbitrary, as no information about the word themselves is taken into account.
Excursion: Skip-gram model

\[ w_{n-1} \quad w_{n-2} \quad w_{n+1} \quad w_{n+2} \]

\[ [\text{Mikolov et al., 2013}] \]
Excursion: Skip-gram model

• Assumption: similar words appear in similar contexts.
• Goal: similar words have similar representations (as they will predict similar contexts).
• Indeed:
  • \( \text{vec}(\text{King}) - \text{vec}(\text{Man}) + \text{vec}(\text{Woman}) \) results in a vector that is closest to \( \text{Queen} \).
  • \( \text{vec}(\text{Madrid}) - \text{vec}(\text{Spain}) + \text{vec}(\text{France}) \) results in a vector that is closest to \( \text{Paris} \).
Excursion: Skip-gram model

Country and Capital Vectors Projected by PCA

- China
- Beijing
- Russia
- Moscow
- Japan
- Tokyo
- Turkey
- Ankara
- Poland
- Warsaw
- Germany
- Berlin
- France
- Paris
- Italy
- Athens
- Spain
- Rome
- Greece
- Madrid
- Portugal
- Lisbon
Excursion: word2vec

Different implementations available

- One of the most well known: word2vec by Mikolov et al.

For machine translation:

- Embeddings trained at the same time as the full system.
- Pre-trained embeddings may be used for initialization.
  - Useful for other tasks, e.g. NLU.
  - No gains reported for MT.

https://code.google.com/archive/p/word2vec/
Recap

- Language model:
  \[ p(w_1^N) \]

- Chain rule: (mathematical equality)
  \[ p(w_1^N) = \prod_{n=1}^{N} p(w_n|w_{1:n-1}) \]

- \( k \)-th order Markov assumption: (\( k + 1 \))-grams
  \[ p(w_1^N) \approx \prod_{n=1}^{N} p(w_n|w_{n-k}^{n-1}) \]
Advantage of NNLMs we encountered up to this point:

- FF language models deal with the sparsity problem (by projecting into a continuous space).
Advantage of NNLMs we encountered up to this point:

- FF language models deal with the sparsity problem (by projecting into a continuous space).
  
  ...but they still are under the Markov chain assumption.
Advantage of NNLMs we encountered up to this point:

- FF language models deal with the sparsity problem (by projecting into a continuous space).
  
  ...but they still are under the Markov chain assumption.

We would like to be able to take into account the whole history.
→ Let the network remember everything it has seen!
Recurrent NNs

\[ p(w) \]

\[ y[t] = f(Wx[t] + Ry[t-1] + b) \]
Recurrent NNs

\[ p(w) \]

In Equations:

\[ y[t] = f(Wx[t] + Ry[t-1] + b) \]
Recurrent NNs

In Equations: $y^{[t]} = f(Wx^{[t]} + Ry^{[t-1]} + b)$
$p(w_1^4) =$
Recurrent NNs

\[ p(w_1^4) = p(w_1 | <s>) \]
Recurrent NNs

\[ p(w_4^4) = \]

\[ p(w_1 | < s >) \]

\[ \times p(w_2 | w_1, < s >) \]
Recurrent NNs

\[ p(w_4^4) = \]
\[ p(w_1 | < s >) \]
\[ \times p(w_2 | w_1, < s >) \]
\[ \times p(w_3 | w_2, w_1, < s >) \]
Recurrent NNs

\[
p(w_4^4) = \\
p(w_1|<s>) \\
\times p(w_2|w_1, <s>) \\
\times p(w_3|w_2, w_1, <s>) \\
\times p(w_4|w_3, w_2, w_1, <s>)
\]
Backpropagation through time

How to train a RNN?

- Use backpropagation.
- Unfold recurrent connections through time.
- Results in a wide network, backpropagation can be used.
- Use chain rule not only for layers, but also for time steps.
Backpropagation through time

\[ x^{[4]} \rightarrow y^{[4]} \rightarrow x^{[4]} \]
Backpropagation through time
Backpropagation through time
Backpropagation through time
Backpropagation through time

\[
\frac{\partial L}{\partial \theta} = m
\]
Backpropagation through time

\[
\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y^{[4]}}
\]
Backpropagation through time

\[
\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y^{[4]}}
\]

\[
\frac{\partial y^{[4]}}{\partial y^{[3]}}
\]

DIAGRAM:
- \(x^{[1]}\) to \(y^{[1]}\)
- \(x^{[2]}\) to \(y^{[2]}\)
- \(x^{[3]}\) to \(y^{[3]}\)
- \(x^{[4]}\)
Backpropagation through time

$$\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y^{[4]}}$$
Backpropagation through time

\[
\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y^{[4]}}
\]

Diagram:

- \( y^{[1]} \) to \( x^{[1]} \)
- \( y^{[2]} \) to \( y^{[1]} \) and \( x^{[2]} \)
- \( y^{[3]} \) to \( y^{[2]} \) and \( x^{[3]} \)
- \( y^{[4]} \) to \( y^{[3]} \) and \( y^{[4]} \) to \( x^{[4]} \)
Backpropagation through time

\[
\frac{\partial L}{\partial \theta} = \frac{\partial L}{\partial y^{[4]}} \frac{\partial y^{[4]}}{\partial y^{[3]}} \frac{\partial y^{[3]}}{\partial y^{[2]}} \frac{\partial y^{[2]}}{\partial y^{[1]}} \frac{\partial y^{[1]}}{\partial \theta}
\]

Diagram showing the flow of partial derivatives through time.
Exploding and vanishing gradient

Observation: sometimes the gradient “misbehaves”.

• Sometimes vanishes ($\text{norm} \approx 0$).
• Sometimes explodes ($\text{norm} \to \infty$).
Exploding and vanishing gradient

Observation: sometimes the gradient “misbehaves”.

- Sometimes vanishes (norm $\approx 0$).
Exploding and vanishing gradient

Observation: sometimes the gradient “misbehaves”.

- Sometimes vanishes (norm $\approx 0$).
- Sometimes explodes (norm $\rightarrow \infty$).
Observation: sometimes the gradient “misbehaves”.

- Sometimes *vanishes* (norm $\approx 0$).
- Sometimes *explodes* (norm $\to \infty$).
Exploding and vanishing gradient

What to do?

- Exploding gradient: clip the gradient (divide by the norm).
  [Full vector or element-wise]
Exploding and vanishing gradient

What to do?

- Exploding gradient: clip the gradient (divide by the norm).
  [Full vector or element-wise]
Exploding and vanishing gradient

What to do?

- Exploding gradient: clip the gradient (divide by the norm).
  [Full vector or element-wise]

- Vanishing gradient: No easy solution.
Exploding and vanishing gradient

Why does this happen?

Sequence of length $T$, $y^{[t]} = f(Wx^{[t]} + Ry^{[t-1]} + b)$. 
Exploding and vanishing gradient

Why does this happen?

Sequence of length $T$, $y^t = f(Wx^t + Ry^{t-1} + b)$.

Derivative of the loss function $\mathcal{L}$:

$$
\frac{\partial \mathcal{L}}{\partial \theta} = \sum_{1 \leq t_2 \leq T} \frac{\partial \mathcal{L}^{[t_2]}}{\partial \theta} = \sum_{1 \leq t_2 \leq T} \sum_{1 \leq t_1 \leq t_2} \frac{\partial \mathcal{L}^{[t_2]}}{\partial y^{[t_2]}} \frac{\partial y^{[t_2]}}{\partial y^{[t_1]}} \frac{\partial y^{[t_1]}}{\partial \theta}
$$

$$
\frac{\partial y^{[t_2]}}{\partial y^{[t_1]}} = \prod_{t_1 < t \leq t_2} \frac{\partial y^{[t]}}{\partial y^{[t-1]}}
$$
Exploding and vanishing gradient

\[
\frac{\partial y^{[t_2]}}{\partial y^{[t_1]}} = \prod_{t_1 < t \leq t_2} \frac{\partial y^{[t]}}{\partial y^{[t-1]}}
\]
Exploding and vanishing gradient

\[
\frac{\partial y[t_2]}{\partial y[t_1]} = \prod_{t_1 < t \leq t_2} \frac{\partial y[t]}{\partial y[t-1]}
\]

It can be shown:

\[
\left\| \frac{\partial y[t]}{\partial y[t-1]} \right\| \leq \|R^T\| \left\| \text{diag} \left( f'(Ry^{t-1}) \right) \right\| \leq \gamma \sigma_{\text{max}}
\]

with

- \(\gamma\) a maximal bound for \(f'(Ry^{t-1})\).
- e.g. \(|\tanh'(x)| \leq 1; |\sigma'(x)| \leq \frac{1}{4}\).
- \(\sigma_{\text{max}}\) the largest singular value of \(R^T\).

[Pascanu et al., 2013] and previous work
LSTMs: Intuition

- RNNs blindly pass information from one state to the other.
- LSTMs include mechanisms for
LSTMs: Intuition

- RNNs blindly pass information from one state to the other.
- LSTMs include mechanisms for
  - Ignoring the input.
LSTMs: Intuition

- RNNs blindly pass information from one state to the other.
- LSTMs include mechanisms for
  - Ignoring the input.
  - Suppressing the “current” output.
LSTMs: Intuition

- RNNs blindly pass information from one state to the other.
- LSTMs include mechanisms for
  - Ignoring the input.
  - Suppressing the “current” output.
  - Forgetting the history.
RNN units

Diagram: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
LSTM units

Diagram: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
LSTM Equations

Compute a “candidate value”, similar to RNNs:

\[ \tilde{C}_t = \tanh(W_c x_t + U_c y_{t-1} + b_c) \]

[Hochreiter and Schmidhuber, 1997]
LSTM Equations

Compute a “candidate value”, similar to RNNs:

\[ \tilde{C}_t = \tanh(W_c x_t + U_c y_{t-1} + b_c) \]

Input gate: control the influence of the current input.

\[ i_t = \sigma(W_i x_t + U_i y_{t-1} + b_i) \]

[Hochreiter and Schmidhuber, 1997]
LSTM Equations

Compute a “candidate value”, similar to RNNs:

\[ \tilde{C}_t = \tanh(W_c x_t + U_c y_{t-1} + b_c) \]

Input gate: control the influence of the current input.

\[ i_t = \sigma(W_i x_t + U_i y_{t-1} + b_i) \]

Forget gate: control the influence of the history.

\[ f_t = \sigma(W_f x_t + U_f y_{t-1} + b_f) \]

[Hochreiter and Schmidhuber, 1997]
LSTM Equations

Memory cell state: combination of new and old state.

\[ C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \]

[Hochreiter and Schmidhuber, 1997]
LSTM Equations

Memory cell state: combination of new and old state.

\[ C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \]

Output gate: how much we want to output to the exterior.

\[ o_t = \sigma(W_o x_t + U_o y_{t-1} + b_o) \]

[Hochreiter and Schmidhuber, 1997]
LSTM Equations

Memory cell state: combination of new and old state.

\[ C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \]

Output gate: how much we want to output to the exterior.

\[ o_t = \sigma(W_o x_t + U_o y_{t-1} + b_o) \]

Output of the cell:

\[ y_t = o_t \odot \tanh(C_t) \]

[Hochreiter and Schmidhuber, 1997]
LSTM Visualization

Compute a “candidate value”, similar to RNNs

Input gate: control the influence of the current output

\[ \tilde{C}_t = \tanh(W_c x_t + U_c y_{t-1} + b_c) \]

\[ i_t = \sigma(W_i x_t + U_i y_{t-1} + b_i) \]
Forget gate: control the influence of the history

\[ f_t = \sigma(W_f x_t + U_f y_{t-1} + b_f) \]
LSTM Visualization

Memory cell state: combination of new and old state

\[ C_t = i_t \odot \tilde{C}_t + f_t \odot C_{t-1} \]
LSTM Visualization

Output gate: how much we want to output to the exterior

Output of the cell

\[
o_t = \sigma(W_o x_t + U_o y_{t-1} + b_o)
\]

\[
y_t = o_t \odot \tanh(C_t)
\]
LSTM Visualization
LSTMs: additional remarks

• LSTMs solve the vanishing gradient problem, but the gradient can still explode.
  • Use gradient clipping.
LSTMs: additional remarks

- LSTMs solve the vanishing gradient problem, but the gradient can still explode.
  - Use gradient clipping.
- Different variants of LSTMs. Basic idea is similar, but
  - Different gates.
  - Different parametrization of the gates.
  - Pay attention when reading the literature.
Gated Recurrent Units:

- Combine forget and input gates into an “update gate”.
- Suppress output gate.
- Add a “reset gate”.

Simpler than LSTMs (less parameters) and similar performance.

\[
\begin{align*}
\mathbf{z}_t &= \sigma(\mathbf{W}_z \mathbf{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z) \\
\mathbf{r}_t &= \sigma(\mathbf{W}_r \mathbf{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r) \\
\tilde{\mathbf{h}}_t &= \tanh(\mathbf{W} \mathbf{x}_t + \mathbf{U}(\mathbf{r}_t \odot \mathbf{h}_{t-1}) + \mathbf{b}) \\
\mathbf{h}_t &= \mathbf{z}_t \odot \tilde{\mathbf{h}}_t + (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1}
\end{align*}
\]

[Cho et al., 2014b]
GRUs Visualization
Neural Machine Translation
The fundamental equation for machine translation

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

is basically a language model expanded with source information.
RNNs give us a way to represent the input.
RNNs give us a way to generate the output.
The encoder creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The encoder creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The encoder creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The encoder creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
The **decoder** generates the translation given the encoder representation.

The **encoder** creates a representation of the input sentence.

[Sutskever et al., 2014, Cho et al., 2014b]
A fixed representation length may not be enough.

Solution: Include an attention mechanism (next lecture).

[Cho et al., 2014a, Bahdanau et al., 2014]
The encoder-decoder allows for great flexibility, e.g.
The encoder-decoder allows for great flexibility, e.g.

- General sequence-to-sequence tasks.
Generalizations

The encoder-decoder allows for great flexibility, e.g.

- General sequence-to-sequence tasks.
- Image based encoder $\rightarrow$ Image captioning system.
The encoder-decoder allows for great flexibility, e.g.

- General sequence-to-sequence tasks.
- Image based encoder $\rightarrow$ Image captioning system.
- Acoustic based encoder $\rightarrow$ Speech translation system.
The encoder-decoder allows for great flexibility, e.g.

- General sequence-to-sequence tasks.
- Image based encoder → Image captioning system.
- Acoustic based encoder → Speech translation system.
- Combination of different encoders → Multimodal translation.
Historical Perspective
Introduce the concept of alignment.

The Commission suggests shorter deadlines, and I agree with this request.

[Brown et al., 1993]
The Commission suggests shorter deadlines, and I agree with this request.

[Coehn et al., 2003, Och and Ney, 2004]
The Commission suggests shorter deadlines, and I agree with this request.
Extract phrases from word alignments.

The Commission suggests shorter deadlines, and I agree with this request.

[Koehn et al., 2003, Och and Ney, 2004]
Log-linear Models

Model the translation probability directly:

\[ p(e_1^I|f_1^J) = \frac{\exp \left( \sum_k \lambda_k f_k(f_1^J, e_1^I) \right)}{\sum_{\hat{e}_1^I} \exp \left( \sum_k \lambda_k f_k(f_1^J, \hat{e}_1^I) \right)} \]

Widely used models:

- Phrase-based models (s2t, t2s).
- Target language model.
- Reordering model.
- Word-based models (s2t, t2s).
- Length heuristics.
- …

[Och and Ney, 2002]
Pyramid of Translation Approaches

interlingua

source text  direct translation  target text

analysis

generation

transfer

[Vauquois, 1968]
## Analysis

<table>
<thead>
<tr>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
</table>

- **Local context**
- **Independent models**
- **Global optimization**
- **Coverage constraints**
- **Over-/under-generation**
- **Model introspection**
- "Black box" approach

**Model size**

- **Misspellings/new words**
<table>
<thead>
<tr>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Local context</td>
<td>+ Global context</td>
</tr>
</tbody>
</table>
Analysis

PBMT

- Local context
  ○ Independent models

NMT

+ Global context
+ Global optimization
Analysis

PBMT

- Local context
  - Independent models
- LM one of many models

NMT

+ Global context
+ Global optimization
+ Generation guided by LM

Coverage constraints
Over-/under-generation
Model introspection
"Black box" approach
Model size
Misspellings/new words
### Analysis

<table>
<thead>
<tr>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Local context</td>
<td>+ Global context</td>
</tr>
<tr>
<td>○ Independent models</td>
<td>+ Global optimization</td>
</tr>
<tr>
<td>- LM one of many models</td>
<td>+ Generation guided by LM</td>
</tr>
<tr>
<td>+ Coverage constraints</td>
<td>- Over-/under-generation</td>
</tr>
</tbody>
</table>

- MT@
<table>
<thead>
<tr>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local context</td>
<td>+ Global context</td>
</tr>
<tr>
<td>Independent models</td>
<td>+ Global optimization</td>
</tr>
<tr>
<td>LM one of many models</td>
<td>+ Generation guided by LM</td>
</tr>
<tr>
<td>Coverage constraints</td>
<td>− Over-/under-generation</td>
</tr>
<tr>
<td>Model introspection</td>
<td></td>
</tr>
</tbody>
</table>
Analysis

**PBMT**
- Local context
  - Independent models
- LM one of many models
+ Coverage constraints
+ Model introspection
- Model size

**NMT**
+ Global context
+ Global optimization
+ Generation guided by LM
- Over-/under-generation
- “Black box” approach
+ Model size
## Analysis

<table>
<thead>
<tr>
<th>PBMT</th>
<th>NMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Local context</td>
<td>+ Global context</td>
</tr>
<tr>
<td>Independent models</td>
<td>+ Global optimization</td>
</tr>
<tr>
<td>LM one of many models</td>
<td>+ Generation guided by LM</td>
</tr>
<tr>
<td>Coverage constraints</td>
<td>- Over-/under-generation</td>
</tr>
<tr>
<td>Model introspection</td>
<td>- “Black box” approach</td>
</tr>
<tr>
<td>Model size</td>
<td>+ Model size</td>
</tr>
<tr>
<td></td>
<td>- Misspellings/new words</td>
</tr>
</tbody>
</table>
Implementation
Deep Learning Toolkits

Efficient algebra (using GPUs) and auto-differentiation.

- MXNet
- Tensorflow
- PyTorch
- Dynet
- [Keras]
- ...

66/70
Implementation of NMT models:

- Sockeye
- OpenNMT
- Marian
- Nematus
- NeuralMonkey
- Tensor2Tensor
- FairSeq
- ...
Conclusions
Conclusions

- Introduced the encoder-decoder architecture.
  - The model presented here does not achieve SOA.
  - But is the base for more advanced models.
- NN allow for integrated modelling and end-to-end training.
- Word embeddings allow to take advantage of word similarities.
The End


The alignment template approach to statistical machine translation.
*Computational Linguistics, Volume 30, Number 4, December 2004.*

On the difficulty of training recurrent neural networks.