Machine Translation 2: Statistical MT: Phrase-Based and Neural

Ondřej Bojar
bojar@ufal.mff.cuni.cz
Institute of Formal and Applied Linguistics
Faculty of Mathematics and Physics
Charles University, Prague

December 2017
MT2: PBMT, NMT
Outline of Lectures on MT

1. Introduction.
   - Why is MT difficult.
   - MT evaluation.
   - Approaches to MT.
   - First peek into phrase-based MT
   - Document, sentence and word alignment.

   - Phrase-based, Hierarchical and Syntactic MT.
   - Neural MT: Sequence-to-sequence.

3. Advanced Topics.
   - Linguistic Features in SMT and NMT.
   - Multilinguality, Multi-Task, Learned Representations.
Outline of MT Lecture 2

1. What makes MT statistical.
   • Brute-force statistical MT.
   • Noisy channeled model.
   • Log-linear model.

2. Phrase-based translation model.
   • Phrase extraction.
   • Decoding (gradual construction of hypotheses).
   • Minimum error-rate training (weight optimization).

3. Neural machine translation (NMT).
   • Sequence-to-sequence, with attention.
Quotes

Warren Weaver (1949):

I have a text in front of me which is written in Russian but I am going to pretend that it is really written in English and that is has been coded in some strange symbols. All I need to do is strip off the code in order to retrieve the information contained in the text.

Noam Chomsky (1969):

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

Frederick Jelinek (80’s; IBM; later JHU and sometimes ÚFAL)

Every time I fire a linguist, the accuracy goes up.

Hermann Ney (RWTH Aachen University):

MT = Linguistic Modelling + Statistical Decision Theory
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.
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 despite translating from English to Czech.
Brute-Force MT (1/2)

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

Good morning. = Dobr´ e ráno.
How are you? = Jak se máš?
How are you? = Jak se máte?

\[ p(e^I_1|f^J_1) = \begin{cases} 
1 & \text{if } e^I_1 = f^J_1 \text{ seen in the TM} \\
0 & \text{otherwise} \end{cases} \] (2)

Any problems with the definition?
Brute-Force MT (2/2)

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?

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

(3)

- Not a probability. There may be \( f^J_1 \), s.t. \( \sum_{e^I_1} p(e^I_1|f^J_1) > 1 \).
  \[
  \Rightarrow \text{ Have to normalize, use } \frac{\text{count}(e^I_1,f^J_1)}{\text{count}(f^J_1)} \text{ instead of 1.}
  \]

- Not “smooth”, no generalization:
  - Good morning.  \( \Rightarrow \)  Dobré ráno.
  - Good evening.  \( \Rightarrow \)  ∅
Bayes’ Law

Bayes’ law for conditional probabilities: $p(a|b) = \frac{p(b|a)p(a)}{p(b)}$

So in our case:

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

$$= \arg\max_{I, e_1} \frac{p(f_1^J|e_1^I)p(e_1^I)}{p(f_1^J)}$$

$$= \arg\max_{I, e_1} p(f_1^J|e_1^I)p(e_1^I)$$

Apply Bayes’ law

$p(f_1^J)$ constant

$\Rightarrow$ irrelevant in maximization

Also called “Noisy Channel” model.
\[
\hat{e}_1^I = \arg\max_{I,e_1^I} p(f_1^J|e_1^I)p(e_1^I)
\]  

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 more reliable.
Without Equations

Parallel Texts

Translation Model

Input

Global Search for sentence with highest probability

Output

Monolingual Texts

Language Model

December 2017

MT2: PBMT, NMT
Summary of Language Models

- $p(e^I_1)$ should report how “good” sentence $e^I_1$ is.
- We surely want $p(\text{The the the.}) < p(\text{Hello.})$
- How about $p(\text{The cat was black.}) < p(\text{Hello.})$?

... We don’t really care in MT. We hope to compare synonymic sentences.

LM is usually a 3-gram language model:

$$
p(\uparrow \uparrow \text{The cat was black} \downarrow \downarrow) = p(\text{The} \uparrow \uparrow \uparrow) p(\text{cat} \uparrow \uparrow \text{The}) p(\text{was} \uparrow \uparrow \text{The cat})$$
$$p(\text{black} \uparrow \downarrow \text{cat was}) p(. \downarrow \uparrow \text{was black}) p(\downarrow \downarrow \text{black} .)
\quad p(\downarrow \downarrow . \downarrow \downarrow)
$$

Formally, with $n = 3$:

$$p_{LM}(e^I_1) = \prod_{i=1}^{I} p(e_i | e^{i-1}_{i-n+1}) \quad (5)$$
Estimating and Smoothing LM

\[ p(w_1) = \frac{\text{count}(w_1)}{\text{total words observed}} \]

Unigram probabilities.

\[ p(w_2|w_1) = \frac{\text{count}(w_1w_2)}{\text{count}(w_1)} \]

Bigram probabilities.

\[ p(w_3|w_2, w_1) = \frac{\text{count}(w_1w_2w_3)}{\text{count}(w_1w_2)} \]

Trigram probabilities.

Unseen ngrams (\( p(\text{ngram}) = 0 \)) are a big problem, invalidate whole sentence: \( p_{\text{LM}}(e_1^I) = \cdots 0 \cdots = 0 \)

\[ \Rightarrow \text{Back-off with shorter ngrams:} \]

\[ p_{\text{LM}}(e_1^I) = \prod_{i=1}^{I} \left( 0.8 \cdot p(e_i|e_{i-1}, e_{i-2}) + \\
0.15 \cdot p(e_i|e_{i-1}) + \\
0.049 \cdot p(e_i) + \\
0.001 \right) \neq 0 \]
Och (2002) discusses some problems of Equation 19:

- Models estimated unreliably ⇒ maybe LM more important:

\[ \hat{e}_1 = \arg\max_{I,e_1^I} p(f_1^J|e_1^I)(p(e_1^I))^2 \]  \hspace{1cm} (7)

- In practice, “direct” translation model equally good:

\[ \hat{e}_1 = \arg\max_{I,e_1^I} p(e_1^I|f_1^J)p(e_1^I) \]  \hspace{1cm} (8)

- Complicated to correctly introduce other dependencies.
  ⇒ Use log-linear model instead.
Log-Linear Model (1)

- $p(e^I_1|f^J_1)$ is modelled as a weighted combination of models, called “feature functions”: $h_1(\cdot, \cdot) \ldots h_M(\cdot, \cdot)$

$$ p(e^I_1|f^J_1) = \frac{\exp(\sum_{m=1}^{M} \lambda_m h_m(e^I_1, f^J_1))}{\sum_{e'^I_1} \exp(\sum_{m=1}^{M} \lambda_m h_m(e'^I_1, f^J_1))} \quad \text{(9)} $$

- Each feature function $h_m(e, f)$ relates source $f$ to target $e$.
  E.g. the feature for $n$-gram language model:

$$ h_{LM}(f^J_1, e^I_1) = \log \prod_{i=1}^{I} p(e_i|e^i_{i-n+1}) \quad \text{(10)} $$

- Model weights $\lambda^M_1$ specify the relative importance of features.
As before, the constant denominator not needed in maximization:

\[
\hat{e}_1^I = \arg\max_{I,e_1^I} \frac{\exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J))}{\sum_{e_1'^I} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1'^I, f_1^J))} \\
= \arg\max_{I,e_1^I} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J))
\]  

(11)
Relation to Noisy Channel

With equal weights and only two features:

- \( h_{TM}(e_1^I, f_1^J) = \log p(f_1^J|e_1^I) \) for the translation model,
- \( h_{LM}(e_1^I, f_1^J) = \log p(e_1^I) \) for the language model,

log-linear model reduces to Noisy Channel:

\[
\hat{e}_1^I = \arg\max_{I, e_1^I} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)) \\
= \arg\max_{I, e_1^I} \exp(h_{TM}(e_1^I, f_1^J) + h_{LM}(e_1^I, f_1^J)) \\
= \arg\max_{I, e_1^I} \exp(\log p(f_1^J|e_1^I) + \log p(e_1^I)) \\
= \arg\max_{I, e_1^I} p(f_1^J|e_1^I)p(e_1^I) \tag{12}
\]
Phrae-based MT: choose such segmentation of input string and such phrase “replacements” to make the output sequence “coherent” (3-grams most probable).
Phrase-Based Translation Model

- Captures the basic assumption of phrase-based MT:
  1. Segment source sentence $f_1^J$ into $K$ phrases $\tilde{f}_1 \ldots \tilde{f}_K$.
  2. Translate each phrase independently: $\tilde{f}_k \rightarrow \tilde{e}_k$.
  3. Concatenate translated phrases (with possible reordering $R$):
     $\tilde{e}_{R(1)} \ldots \tilde{e}_{R(K)}$

- In theory, the segmentation $s_1^K$ is a hidden variable in the maximization, we should be summing over all segmentations: (Note the three args in $h_m(\cdot, \cdot, \cdot)$ now.)

$$\hat{e}_1^I = \text{argmax}_{I,e_1^I} \sum_{s_1^K} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, s_1^K))$$  \hspace{1cm} (13)

- In practice, the sum is approximated with a max (the biggest element only):

$$\hat{e}_1^I = \text{argmax}_{I,e_1^I} \max_{s_1^K} \exp(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J, s_1^K))$$  \hspace{1cm} (14)
Core Feature: Phrase Trans. Prob.

The most important feature: phrase-to-phrase translation:

$$h_{\text{Phr}}(f^J_1, e^I_1, s^K_1) = \log \prod_{k=1}^{K} p(\tilde{f}_k | \tilde{e}_k)$$  \hspace{1cm} (15)

The conditional probability of phrase $\tilde{f}_k$ given phrase $\tilde{e}_k$ is estimated from relative frequencies:

$$p(\tilde{f}_k | \tilde{e}_k) = \frac{\text{count}(\tilde{f}, \tilde{e})}{\text{count}(\tilde{e})}$$  \hspace{1cm} (16)

- $\text{count}(\tilde{f}, \tilde{e})$ is the number of co-occurrences of a phrase pair $(\tilde{f}, \tilde{e})$ that are consistent with the word alignment
- $\text{count}(\tilde{e})$ is the number of occurrences of the target phrase $\tilde{e}$ in the training corpus.
- $h_{\text{Phr}}$ is usually used twice, in both directions: $p(\tilde{f}_k | \tilde{e}_k)$ and $p(\tilde{e}_k | \tilde{f}_k)$
Phrase-Based Features in Moses

Given parallel training corpus, phrases are extracted and scored:

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Score (in europa)</th>
<th>Score (in europe)</th>
</tr>
</thead>
<tbody>
<tr>
<td>in europa</td>
<td></td>
<td></td>
</tr>
<tr>
<td>europas</td>
<td></td>
<td></td>
</tr>
<tr>
<td>in der europaeischen union</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The scores are: \( \phi(\cdot) = \log p(\cdot) \)

- **phrase translation probabilities:** \( \phi_{\text{phr}}(f|e) \) and \( \phi_{\text{phr}}(e|f) \)
- **lexical weighting:** \( \phi_{\text{lex}}(f|e) \) and \( \phi_{\text{lex}}(e|f) \) (Koehn, 2003)

\[
\phi_{\text{lex}}(f|e) = \log \max_{a \in \text{alignments of } (f,e)} \prod_{i=1}^{\frac{|f|}{|e|}} \frac{1}{\{j \mid (i,j) \in a\}} \sum_{\forall (i,j) \in a} p(f_i|e_j)
\]  

(17)
Other Features Used in PBMT

- Word count/penalty: \( h_{wp}(e_1^I, \cdot, \cdot) = I \)
  \( \Rightarrow \) Do we prefer longer or shorter output?

- Phrase count/penalty: \( h_{pp}(\cdot, \cdot, s^K_1) = K \)
  \( \Rightarrow \) Do we prefer translation in more or fewer less-dependent bits?

- Reordering model: different basic strategies (Lopez, 2009)
  \( \Rightarrow \) Which source spans can provide continuation at a moment?

- \( n \)-gram LM:

  \[
  h_{LM}(\cdot, e_1^I, \cdot) = \log \prod_{i=1}^I p(e_i | e_{i-n+1}^{i-1})
  \]  
  (18)

  \( \Rightarrow \) Is output \( n \)-gram-wise coherent?
See slides by Philipp Koehn.

- Creating translation options.
- Expanding hypotheses.
- Recombining hypotheses.
- Stack-based pruning.
Local and Non-Local Features

- Local features decompose along hypothesis construction.
  - Phrase- and word-based features.
- Non-local features span the boundaries (e.g. LM).

December 2017  MT2: PBMT, NMT
Weight Optimization: MERT Loop

Minimum Error Rate Training (Och, 2003)
- Higher phrase penalty chops sentence into more segments.
- Too strong LM weight leads to words dropped.
- Negative LM weight leads to obscure wordings.

December 2017  MT2: PBMT, NMT
Summary of PBMT

Phrase-based MT:

• is a log-linear model
• assumes phrases relatively independent of each other
• decomposes sentence into contiguous phrases
• search has two parts:
  – lookup of all relevant translation options
  – stack-based beam search, gradually expanding hypotheses

To train a PBMT system:

1. Align words.
2. Extract (and score) phrases consistent with word alignment.
3. Optimize weights (MERT).
1: Align Training Sentences

Nemám žádného psa.   Viděl kočku.
I have no dog.       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.
New input: Nemám žádného psa.

New input: Viděl kočku.

... I don't have cat.
Nemám žádného psa. (I have no dog.)

Viděl kočku. (He saw a cat.)

New input: Nemám kočku. (I have no cat.)
Nemám žádného psa.
I have no dog.

Viděl kočku.
He saw a cat.

New input:
Nemám kočku.
I have a cat.

... I don't have cat.
Meaning Got Reversed!

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

Nemám žádného psa. I have no dog.

Viděl kočku. He saw a cat.
What Went Wrong?

\[ \hat{e}_1^I = \arg\max_{I, e_1^I} p(f_1^J|e_1^I)p(e_1^I) = \arg\max_{I, e_1^I} \prod_{(\hat{f}, \hat{e}) \in \text{phrase pairs of } f_1^J, e_1^I} p(\hat{f}|\hat{e})p(e_1^I) \] (19)

- **Too strong phrase-independence assumption.**
  - 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.
- **Language model is a separate unit.**
  - \( p(e_1^I) \) models the target sentence independently of \( f_1^J \).
Redefining $p(e^I_1|f^J_1)$

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

$$p(e^I_1|f^J_1) = p(e_1, e_2, \ldots e_I|f^J_1)$$

$$= p(e_1|f^J_1) \cdot p(e_2|e_1, f^J_1) \cdot p(e_3|e_2, e_1, f^J_1) \ldots$$

$$= \prod_{i=1}^{I} p(e_i|e_1, \ldots e_{i-1}, f^J_1)$$

(20)

... 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?
NNs: Universal Approximators

- A neural network with a single hidden layer (possibly huge) can approximate any continuous function to any precision.
- (Nothing claimed about learnability.)

Bad Features & Low Depth

Test loss 0.510
Training loss 0.488

X1
X2
X12
X22
X1X2
Too Complex NN Fails to Learn

Test loss 0.195
Training loss 0.208

This is the output from one neuron. Hover to see it larger.

The outputs are mixed with varying weights, shown by the thickness of the lines.
Deep NNs for Image Classification

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

Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]
Representation Learning

• Based on training data (sample inputs and expected outputs)
• the neural network learns by itself
• what is important in the inputs
• to predict the outputs best.

A “representation” is a new set of axes.

• Instead of 3 dimensions \((x, y, \text{color})\), we get
• 2000 dimensions: \((\text{elephantity}, \text{number of storks}, \text{blueness}, \ldots)\)
• designed automatically to help in best prediction of the output
One Layer \( \tanh(Wx + b) \), 2D→2D

Skew: \\
\( W \)

Transpose: \\
\( b \)

Non-lin.: \\
\( \tanh \)

Four Layers, Disentangling Spirals


December 2017  MT2: PBMT, NMT
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:

  \[
  \begin{array}{ccccccc}
  & \text{the} & \text{cat} & \text{is} & \text{on} & \text{the} & \text{mat} \\
  \hline
  \uparrow & a & 0 & 0 & 0 & 0 & 0 & 0 \\
  & about & 0 & 0 & 0 & 0 & 0 & 0 \\
  & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
  & \text{cat} & 0 & 1 & 0 & 0 & 0 & 0 \\
  & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
  & \text{is} & 0 & 0 & 1 & 0 & 0 & 0 \\
  & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
  & \text{on} & 0 & 0 & 0 & 1 & 0 & 0 \\
  & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
  & \text{the} & 1 & 0 & 0 & 0 & 1 & 0 \\
  & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\
  \downarrow & \text{zebra} & 0 & 0 & 0 & 0 & 0 & 0 \\
  \end{array}
  \]

Main drawback: No relations, all words equally close/far.

December 2017

MT2: PBMT, NMT
### Processing Text with NNs

- **Map each word to a vector of 0s and 1s ("1-hot repr.")**:
  
  $\text{cat} \rightarrow (0, 0, \ldots, 0, 1, 0, \ldots, 0)$

- **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>on</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</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.

December 2017

MT2: PBMT, NMT
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>\uparrow 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>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</td>
<td>\ldots</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: \ldots \ldots \ldots \ldots \ldots \ldots

1.3M English
2.2M Czech
```

- Main drawback: No relations, all words equally close/far.

December 2017

MT2: PBMT, NMT
Word Embeddings

- Map each word to a dense vector.
- In practice 300–2000 dimensions are used, not 1–2M.
  - The dimensions have no clear interpretation.
- Embeddings are trained for each particular task.
  - NNs: The matrix that maps 1-hot input to the first layer.
- The famous word2vec (Mikolov et al., 2013):
  - CBOW: Predict the word from its four neighbours.
  - Skip-gram: Predict likely neighbours given the word.

Right: CBOW with just a single-word context (http://www-personal.umich.edu/~ronxin/pdf/w2vexp.pdf)

December 2017  MT2: PBMT, NMT
Continuous Space of Words

Word2vec embeddings show interesting properties:

\[ v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen}) \]  

Illustrations from https://www.tensorflow.org/tutorials/word2vec

December 2017    MT2: PBMT, NMT
Variable-Length Inputs

Variable-length input can be handled by recurrent NNs:

- Reading one input symbol at a time.
  - The same (trained) transformation used every time.
- Unroll in time (up to a fixed length limit).

Tricks needed to train (to avoid “vanishing gradients”):

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

\[
\begin{align*}
    z_t &= \sigma (W_z \cdot [h_{t-1}, x_t]) \\
    r_t &= \sigma (W_r \cdot [h_{t-1}, x_t]) \\
    \tilde{h}_t &= \tanh (W \cdot [r_t \cdot h_{t-1}, x_t]) \\
    h_t &= (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t
\end{align*}
\]
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
<table>
<thead>
<tr>
<th>Syntactic Similarity (“of the”)</th>
</tr>
</thead>
<tbody>
<tr>
<td>the launch of the</td>
</tr>
<tr>
<td>the fall of the</td>
</tr>
<tr>
<td>the edge of the</td>
</tr>
<tr>
<td>the capital of the</td>
</tr>
<tr>
<td>the president of the</td>
</tr>
<tr>
<td>the records of the</td>
</tr>
<tr>
<td>the people</td>
</tr>
<tr>
<td>the evolution of the</td>
</tr>
<tr>
<td>the scene of the</td>
</tr>
<tr>
<td>the members of the</td>
</tr>
<tr>
<td>the majority of the</td>
</tr>
<tr>
<td>the back of the</td>
</tr>
<tr>
<td>the valley of the</td>
</tr>
<tr>
<td>the Chairperson of the</td>
</tr>
<tr>
<td>the portion of the</td>
</tr>
<tr>
<td>the approval of the</td>
</tr>
<tr>
<td>the application of the</td>
</tr>
<tr>
<td>the resident of the</td>
</tr>
<tr>
<td>the sister of the</td>
</tr>
<tr>
<td>the Ministry of the</td>
</tr>
<tr>
<td>the work of the</td>
</tr>
<tr>
<td>the origin of the</td>
</tr>
<tr>
<td>the authors of the</td>
</tr>
<tr>
<td>the peak of the</td>
</tr>
<tr>
<td>the release of the</td>
</tr>
<tr>
<td>the establishment of the</td>
</tr>
<tr>
<td>the voices of the</td>
</tr>
<tr>
<td>the power of the</td>
</tr>
<tr>
<td>the Director of the</td>
</tr>
<tr>
<td>the ice of the</td>
</tr>
<tr>
<td>the decision of the</td>
</tr>
<tr>
<td>the risk of the</td>
</tr>
<tr>
<td>the capital of the</td>
</tr>
<tr>
<td>the artists of the</td>
</tr>
<tr>
<td>the special role of the</td>
</tr>
<tr>
<td>the Museum</td>
</tr>
<tr>
<td>the CEO of the</td>
</tr>
<tr>
<td>the restructuring of the</td>
</tr>
<tr>
<td>the presence of the</td>
</tr>
<tr>
<td>the absence of the</td>
</tr>
<tr>
<td>the beauty of the</td>
</tr>
<tr>
<td>the winner of the</td>
</tr>
<tr>
<td>the head of the</td>
</tr>
<tr>
<td>as the head of the</td>
</tr>
<tr>
<td>the Go</td>
</tr>
</tbody>
</table>

December 2017 | MT2: PBMT, NMT | 54
Semantic Similarity (Countries)

December 2017
MT2: PBMT, NMT
Sub-Word Units

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

⇒ 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>čes ký □ politik □ s vez l □ migrant y</td>
</tr>
<tr>
<td>BPE 30k</td>
<td>český politik s@@ vez@@ l mi@@ granty</td>
</tr>
</tbody>
</table>

BPE (Byte-Pair Encoding) uses $n$ most common substrings (incl. frequent words).
Sutskever et al. (2014) use:

- LSTM RNN encoder-decoder
- to consume and produce variable-length sentences.

First the Encoder:

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

Remember: \[ p(e_1^I | f_1^J) = p(e_1 | f_1^J) \cdot p(e_2 | e_1, f_1^J) \cdot p(e_3 | e_2, e_1, f_1^J) \ldots \]

- Again RNN, producing one word at a time.
- The produced word fed back into the network.
  - (Word embeddings in the target language used here.)
Encoder-Decoder Architecture

$f = (La, \text{croissance, } \text{économique, } s'est, \text{ralentie, } \text{ces, dernières, } \text{années, } , )$

$\epsilon = (\text{Economic, growth, has, slowed, down, in, recent, years, } , )$

Continuous Space of Sentences

2-D PCA projection of 8000-D space representing sentences (Sutskever et al., 2014).

I gave her a card in the garden
In the garden, I gave her a card
She was given a card by me in the garden
In the garden, she gave me a card
I was given a card by her in the garden
She gave me a card in the garden
She was given a card by me in the garden
In the garden, I gave her a card
I gave her a card in the garden
Attention (1/2)

- Arbitrary-length sentences fit badly into a fixed vector.
- Reading input backward works better.

... because early words will be more salient.

⇒ Use Bi-directional RNN and “attend” to all states $h_i$. 
Attention (2/2)

- Add a sub-network predicting importance of source states at each step.
Attention $\approx$ Alignment

- We can collect the attention across time.
- Each column corresponds to one decoder time step.
- Source tokens correspond to rows.
Ultimate Goal of SMT vs. NMT

Goal of “classical” SMT:

Find \textit{minimum translation units} \sim graph partitions:

- such that they are frequent across many sentence pairs.
- without imposing (too hard) constraints on reordering.
- in an unsupervised fashion.

Goal of neural MT:

\textbf{Avoid} minimum translation units. Find NN architecture that

- Reads input in as original form as possible.
- Produces output in as final form as possible.
- Can be optimized end-to-end \textit{in practice}.
Is NMT That Much Better?

The outputs of this year’s best system: http://matrix.statmt.org/

SRC A 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week.

Osmadvacetiletý kuchař, který se nedávno přestěhoval do San Francisc, byl tento týden nalezen mrtvý na schodišti místního obchodního centra.

Osmadvacetiletý šéfkuchař, který se nedávno přistěhoval do San Franciska, byl tento týden Ø schodech místního obchodu.

SRC There were creative differences on the set and a disagreement.

Došlo ke vzniku kreativních rozdílů na scéně a k neshodám.
Na place byly tvůrčí rozdíly a neshody.
Is NMT That Much Better?

The outputs of this year’s best system: http://matrix.statmt.org/

SRC  A 28-year-old chef who had recently moved to San Francisco was found dead in the stairwell of a local mall this week.

MT   Osmadvacetiletý kuchař, který se nedávno přestěhoval do San Francisca, byl tento týden nalezen mrtvý na schodišti místního obchodního centra.

REF  Osmadvacetiletý šéfkuchař, který se nedávno přistěhoval do San Franciska, byl tento týden řízen na schodech místního obchodu.

SRC  There were creative differences on the set and a disagreement.

REF  Došlo ke vzniku kreativních rozdílů na scéně a k neshodám.

MT   Na place byly tvůrčí rozdíly a neshody.
Luckily ;-) Bad Errors Happen

SRC  ... said Frank initially stayed in hostels...
MT   ... řekl, že Frank původně zůstal v Budějovicích...

SRC  Most of the Clintons’ income...
MT   Většinu příjmů Kliniky...

SRC  The 63-year-old has now been made a special representative...
MT   63letý mladík se nyní stal zvláštním zástupcem...

SRC  He listened to the moving stories of the women.
MT   Naslouchal pohyblivým příběhům žen.
Catastrophic Errors

SRC Criminal Minds star Thomas Gibson sacked after hitting producer

REF Thomas Gibson, hvězda seriálu Myšlenky zločince, byl propuštěn po té, co uhodil režiséra

MT Kriminalisté Minsku hvězdu Thomase Gibsona vyhostili po zásahu producenta

SRC ...add to that its long-standing grudge...

REF ...přidejte k tomu svou dlouholetou nenávist...

MT ...přidejte k tomu svou dlouholetou záštitu...
(grudge → zášť → záštita)
German→Czech SMT vs. NMT

- A smaller dataset, very first (but comparable) results.
- NMT performs better on average, but occasionally:

| SRC | Das Spektakel ähnelt dem Eurovision Song Contest. |
| REF | Je to jako pěvecká soutěž Eurovision. |
| SMT | Podívanou připomíná hudební soutěž Eurovize. |
| NMT | Divadlo se podobá Eurovizi Conview. |

| SRC | Erderwärmung oder Zusammenstoß mit Killerasteroid. |
| REF | Globální oteplení nebo kolize se zabijáckým asteroidem. |
| SMT | Globální oteplování, nebo srážka s Killerasteroid. |
| NMT | Globální oteplování, nebo střet s zabijákem. |

| SRC | Zu viele verletzte Gefühle. |
| REF | Příliš mnoho nepřátelských pocitů. |
| SMT | Příliš mnoho zraněných pocity. |
| NMT | Příliš mnoho zraněných. |
Two crucially different models covered:

- Phrase-based: contiguous but independent phrases.
- Neural: unit-less, continuous space.
References


Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. pages 3104–3112.