Machine Translation
07: Open-vocabulary Translation

Rico Sennrich

University of Edinburgh
1 Refresher

2 Open-vocabulary models
   - Non-Solution: Ignore Rare Words
   - Solution 1: Approximative Softmax
   - Solution 2: Back-off Models
   - Solution 3: Subword NMT
   - Solution 4: Character-level NMT
how do we represent text in NMT?

- 1-hot encoding
  - lookup of word embedding for input
  - probability distribution over vocabulary for output

- large vocabularies
  - increase network size
  - decrease training and decoding speed

- typical network vocabulary size: 10,000–100,000 symbols

<table>
<thead>
<tr>
<th>vocabulary</th>
<th>representation of &quot;cat&quot;</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>the 1-hot vector</td>
</tr>
<tr>
<td>1</td>
<td>cat</td>
</tr>
<tr>
<td>2</td>
<td>is</td>
</tr>
<tr>
<td>.</td>
<td>.</td>
</tr>
<tr>
<td>1024</td>
<td>mat</td>
</tr>
</tbody>
</table>
Problem

translation is open-vocabulary problem

- many training corpora contain millions of word types
- productive word formation processes (compounding; derivation) allow formation and understanding of unseen words
- names, numbers are morphologically simple, but open word classes
Non-Solution: Ignore Rare Words

- replace out-of-vocabulary words with UNK
- a vocabulary of 50,000 words covers 95% of text

this gets you 95% of the way...
... if you only care about automatic metrics
Non-Solution: Ignore Rare Words

- replace out-of-vocabulary words with UNK
- a vocabulary of 50,000 words covers 95% of text

this gets you 95% of the way...
... if you only care about automatic metrics

why 95% is not enough

rare outcomes have high self-information

<table>
<thead>
<tr>
<th>source reference</th>
<th>The indoor temperature is very pleasant. Das Raumklima ist sehr angenehm.</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Bahdanau et al., 2015]</td>
<td>Die UNK ist sehr angenehm.</td>
</tr>
<tr>
<td>[Jean et al., 2015]</td>
<td>Die Innenpool ist sehr angenehm.</td>
</tr>
</tbody>
</table>
**Solution 1: Approximative Softmax**

**approximative softmax [Jean et al., 2015]**

- compute softmax over "active" subset of vocabulary
  - smaller weight matrix, faster softmax
  - at training time: vocabulary based on words occurring in training set partition
  - at test time: determine likely target words based on source text (using cheap method like translation dictionary)

**limitations**

- allows larger vocabulary, but still not open
- network may not learn good representation of rare words
Solution 2: Back-off Models

back-off models [Jean et al., 2015, Luong et al., 2015]
- replace rare words with UNK at training time
- when system produces UNK, align UNK to source word, and translate this with back-off method

source: The indoor temperature is very pleasant.
reference: Das Raumklima ist sehr angenehm.

limitations
- compounds: hard to model 1-to-many relationships
- morphology: hard to predict inflection with back-off dictionary
- names: if alphabets differ, we need transliteration
- alignment: attention model unreliable

<table>
<thead>
<tr>
<th></th>
<th>target language</th>
<th>source language</th>
<th>bilingual model</th>
<th>translation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bahdanau et al., 2015</td>
<td>Die UNK ist sehr angenehm.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Jean et al., 2015</td>
<td>Die Innenpool ist sehr angenehm.</td>
<td>X</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Refresher

Open-vocabulary models
- Non-Solution: Ignore Rare Words
- Solution 1: Approximative Softmax
- Solution 2: Back-off Models
- Solution 3: Subword NMT
- Solution 4: Character-level NMT
Subwords for NMT: Motivation

MT is an open-vocabulary problem

- compounding and other productive morphological processes
  - they charge a carry-on bag fee.
  - sie erheben eine Handgepäckgebühr.
- names
  - Obama (English; German)
  - Обама (Russian)
  - オバマ (o-ba-ma) (Japanese)
- technical terms, numbers, etc.
Subword units

segmentation algorithms: wishlist

- **open-vocabulary NMT**: encode *all* words through small vocabulary
- encoding generalizes to unseen words
- small text size
- good translation quality

our experiments [Sennrich et al., 2016]

- after preliminary experiments, we propose:
  - character n-grams (with shortlist of unsegmented words)
  - segmentation via *byte pair encoding* (BPE)
Byte pair encoding for word segmentation

bottom-up character merging

- starting point: character-level representation → computationally expensive
- compress representation based on information theory → byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair (’A’,’B’) with ’AB’
- hyperparameter: when to stop → controls vocabulary size

<table>
<thead>
<tr>
<th>word</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>’l o w&lt;/w&gt;’</td>
<td>5</td>
</tr>
<tr>
<td>’l o w e r&lt;/w&gt;’</td>
<td>2</td>
</tr>
<tr>
<td>’n e w e s t&lt;/w&gt;’</td>
<td>6</td>
</tr>
<tr>
<td>’w i d e s t&lt;/w&gt;’</td>
<td>3</td>
</tr>
</tbody>
</table>

vocabulary:

l o w</w> w e r</w> n s t</w> i d
Byte pair encoding for word segmentation

**bottom-up character merging**

- starting point: character-level representation → computationally expensive
- compress representation based on information theory → byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair (’A’, ’B’) with ’AB’
- hyperparameter: when to stop → controls vocabulary size

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

bottom-up character merging

- starting point: character-level representation → computationally expensive
- compress representation based on information theory → byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair (’A’,’B’) with ’AB’
- hyperparameter: when to stop → controls vocabulary size

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

bottom-up character merging

- starting point: character-level representation  
  → computationally expensive
- compress representation based on information theory  
  → byte pair encoding [Gage, 1994]
- repeatedly replace most frequent symbol pair (’A’,’B’) with ’AB’
- hyperparameter: when to stop  
  → controls vocabulary size

<table>
<thead>
<tr>
<th>word</th>
<th>freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>’lo w&lt;/w&gt;’</td>
<td>5</td>
</tr>
<tr>
<td>’lo w e r&lt;/w&gt;’</td>
<td>2</td>
</tr>
<tr>
<td>’n e w est&lt;/w&gt;’</td>
<td>6</td>
</tr>
<tr>
<td>’w i d est&lt;/w&gt;’</td>
<td>3</td>
</tr>
</tbody>
</table>

vocabulary:

lo w</w> w e r</w> n s t</w> i d es est</w> lo
why BPE?

- open-vocabulary: operations learned on training set can be applied to unknown words
- compression of frequent character sequences improves efficiency → trade-off between text length and vocabulary size

'l o w e s t</w>'

<table>
<thead>
<tr>
<th>before</th>
<th>after</th>
</tr>
</thead>
<tbody>
<tr>
<td>'l o w e s t&lt;/w&gt;'</td>
<td>es</td>
</tr>
<tr>
<td>es t&lt;/w&gt;</td>
<td>est&lt;/w&gt;</td>
</tr>
<tr>
<td>l o</td>
<td>lo</td>
</tr>
</tbody>
</table>
why BPE?

- open-vocabulary: operations learned on training set can be applied to unknown words
- compression of frequent character sequences improves efficiency
  → trade-off between text length and vocabulary size

`'l o w es t<w>'` → `es` → `est<w>`
`l o` → `lo`
why BPE?

- open-vocabulary: operations learned on training set can be applied to unknown words
- compression of frequent character sequences improves efficiency → trade-off between text length and vocabulary size

```plaintext
'low est</w>'

es -> es

es t</w> -> est</w>

lo -> lo
```
Byte pair encoding for word segmentation

why BPE?

- open-vocabulary: operations learned on training set can be applied to unknown words
- compression of frequent character sequences improves efficiency → trade-off between text length and vocabulary size

'lo w est</w>’

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>es</td>
<td>→</td>
</tr>
<tr>
<td>es t</td>
<td>→</td>
</tr>
<tr>
<td>l o</td>
<td>→</td>
</tr>
</tbody>
</table>

es → es
es t → est
lo → lo
## Evaluation: data and methods

### Data

- WMT 15 English → German and English → Russian

### Model

- Attentional encoder–decoder neural network
- Parameters and settings as in [Bahdanau et al, 2014]
Subword NMT: Translation Quality

![BLEU Scores for EN-DE and EN-RU]

- **EN-DE**
  - Word-level NMT (with back-off): 22.0
  - Subword-level NMT: BPE: 22.8

- **EN-RU**
  - Word-level NMT (with back-off): 19.1
  - Subword-level NMT: BPE: 20.4

[Jean et al., 2015]
Subword NMT: Translation Quality

NMT Results EN-RU

unigram $F_1$ vs. training set frequency rank

- subword-level NMT: BPE
- subword-level NMT: char bigrams
- word-level (with back-off)
- word-level (no back-off)
<table>
<thead>
<tr>
<th>system</th>
<th>sentence</th>
</tr>
</thead>
<tbody>
<tr>
<td>source reference</td>
<td>health research institutes</td>
</tr>
<tr>
<td>Gesundheitsforschungsinstitute</td>
<td>Forschungsinstitute</td>
</tr>
<tr>
<td>word-level (with back-off)</td>
<td>Forschungsinstitute</td>
</tr>
<tr>
<td>character bigrams</td>
<td>Gesundheitsforschungsinstitute</td>
</tr>
<tr>
<td>BPE</td>
<td></td>
</tr>
<tr>
<td>source reference</td>
<td>rakfisk</td>
</tr>
<tr>
<td>reference</td>
<td>ракфиска (rakfiska)</td>
</tr>
<tr>
<td>word-level (with back-off)</td>
<td>ракфиск → UNK → ракфиск</td>
</tr>
<tr>
<td>character bigrams</td>
<td>ракфиск → ра́кф́иск (ракфиск)</td>
</tr>
<tr>
<td>BPE</td>
<td>ракфиск → ракфиска (ракфиска)</td>
</tr>
</tbody>
</table>
Open-vocabulary models

- Non-Solution: Ignore Rare Words
- Solution 1: Approximative Softmax
- Solution 2: Back-off Models
- Solution 3: Subword NMT
- Solution 4: Character-level NMT
Character-level Models

- **advantages:**
  - (mostly) open-vocabulary
  - no heuristic or language-specific segmentation
  - neural network can conceivably learn from raw character sequences

- **drawbacks:**
  - increasing sequence length slows training/decoding
    (reported x2–x4 increase in training time)
  - naive char-level encoder-decoders are currently resource-limited
    [Luong and Manning, 2016]

- **open questions**
  - on which level should we represent meaning?
  - on which level should attention operate?
Character-level Models

Hierarchical model: back-off revisited [Luong and Manning, 2016]

- word-level model produces UNKs
- for each UNK, character-level model predicts word based on word hidden state
- pros:
  - prediction is more flexible than dictionary look-up
  - more efficient than pure character-level translation
- cons:
  - independence assumptions between main model and backoff model
6 Quantitative Analysis

Slower Layer for Alignment

On En-De, we test which layer of the decoder should be used for computing soft-alignments. In the case of subword-level decoder, we observed no difference between choosing any of the two layers of the decoder against using the concatenation of all the layers (Table 1 (a–b)). On the other hand, with the character-level decoder, we noticed an improvement when only the slower layer ($h_2$) was used for the soft-alignment mechanism (Table 1 (c–g)). This suggests that the soft-alignment mechanism benefits by aligning a larger chunk in the target with a subword unit in the source, and we use only the slower layer for all the other language pairs.

Single Models

In Table 1, we present a comprehensive report of the translation qualities of (1) subword-level decoder, (2) character-level base decoder and (3) character-level bi-scale decoder, for all the language pairs. We see that the both types of character-level decoder outperform the subword-level decoder for En-Cs and En-Fi quite significantly. On En-De, the character-level base decoder outperforms both the subword-level decoder and the character-level bi-scale decoder, validating the effectiveness of the character-level modelling. On En-Ru, among the single models, the character-level decoders outperform the subword-level decoder, but in general, we observe that all the three alternatives work comparable to each other.

These results clearly suggest that it is indeed possible to do character-level translation without explicit segmentation. In fact, what we observed is that character-level translation often surpasses the translation quality of word-level translation. Of course, we note once again that our experiment is restricted to using an unsegmented character sequence at the decoder only, and a further exploration toward replacing the source sentence with an unsegmented character sequence is needed.

7 Qualitative Analysis

(1) Can the character-level decoder generate a long, coherent sentence? The translation in characters is dramatically longer than that in words, likely making it more difficult for a recurrent neural network to generate a coherent sentence in characters. This belief turned out to be false. As shown in Fig. 2 (left), there is no significant difference between the subword-level and character-level decoders, even though the lengths of the generated translations are generally 5–10 times longer in characters.

(2) Does the character-level decoder help with rare words? One advantage of character-level modelling is that it can model the composition of any character sequence, thereby better modelling rare morphological variants. We empirically confirm this by observing the growing gap in the average negative log-probability of words between the subword-level and character-level decoders as the frequency of the words decreases. This is shown in Fig. 2 (right) and explains one potential cause behind the success of character-level decoding in our experiments (we define $\text{diff}(x,y) = x - y$).

(3) Can the character-level decoder soft-align between a source word and a target character? In Fig. 3 (left), we show an example soft-alignment of a source sentence, “Two sets of light so close to one another”. It is clear that the character-level translation model well captured the alignment between the source subwords and target characters.

Character-level Models

Character-level output [Chung et al., 2016]

- no word segmentation on target side
- encoder is BPE-level
- good results for EN→{DE,CS,RU,FI}
- long training time (∼x2 compared to BPE-level model)
Character-level Models

character-level input [Ling et al., 2015]

hierarchical representation: RNN states represent words, but their representation is computed from character-level LSTM

* C2W Compositional Model

BLSTM

Word Vector for "Where"
Fully Character-level NMT [Lee et al., 2016]

- goal: get rid of word boundaries
- character-level RNN on target side
- source side: convolution and max-pooling layers

Figure 1: Encoder architecture schematics. Underscore denotes padding. A dotted vertical line delimits each segment.

3.3 Challenges

Sentences are on average 6 (DE, CS and RU) to 8 (FI) times longer when represented in characters. This poses three major challenges to achieving fully character-level translation.

1. Training/decoding latency
   - For the decoder, although the sequence to be generated is much longer, each character-level softmax operation costs considerably less compared to a word- or subword-level softmax. Chung et al. (2016) report that character-level decoding is only 14% slower than subword-level decoding.
   - On the other hand, computational complexity of the attention mechanism grows quadratically with respect to the sentence length, as it needs to attend to every source token for every target token. This makes a naive character-level approach, such as in (Luong and Manning, 2016), computationally prohibitive. Consequently, reducing the length of the source sequence is key to ensuring reasonable speed in both training and decoding.

2. Mapping character sequence to continuous representation
   - The arbitrary relationship between the orthography of a word and its meaning is a well-known problem in linguistics (de Saussure, 1916). Building a character-level encoder is arguably a more difficult problem, as the encoder needs to learn a highly non-linear function from a long sequence of character symbols to a meaning representation.

3. Long range dependencies in characters
   - A character-level encoder needs to model dependencies over longer timespans than a word-level encoder does.

4. Fully Character-Level NMT

4.1 Encoder

We design an encoder that addresses all the challenges discussed above by using convolutional and pooling layers aggressively to both (1) drastically shorten the input sentence and (2) efficiently capture local regularities. Inspired by the character-level language model from (Kim et al., 2015), our encoder first reduces the source sentence length with a series of convolutional, pooling and highway layers. The shorter representation, instead of the full character sequence, is passed through a bidirectional GRU to (3) help it resolve long term dependencies.

We illustrate the proposed encoder in Figure 1 and discuss each layer in detail below.

Embedding

We map the source sentence \((x_1, \ldots, x_T) \in \mathbb{R}^1 \times T\) to a sequence of character embeddings \(X = (C(x_1), \ldots, C(x_T)) \in \mathbb{R}^d_c \times T\) where \(C\) is the character embedding lookup table: \(C \in \mathbb{R}^{d_c \times |C|}\).
BPE-level subword segmentation is currently the most widely used technique for open-vocabulary NMT.

Character-level models are theoretically attractive, but currently require specialized architectures and more computational resources.

The presented methods allow open vocabulary; how well we generalize is another question.

→ Next lecture: morphology
Neural Machine Translation by Jointly Learning to Align and Translate.

CoRR, abs/1603.06147.

A New Algorithm for Data Compression.

On Using Very Large Target Vocabulary for Neural Machine Translation.
In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 1–10, Beijing, China. Association for Computational Linguistics.

Fully Character-Level Neural Machine Translation without Explicit Segmentation.
ArXiv e-prints.

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

Addressing the Rare Word Problem in Neural Machine Translation.
In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, pages 11–19, Beijing, China. Association for Computational Linguistics.

Neural Machine Translation of Rare Words with Subword Units.