COS 584

Spring 2021

P9: Machine Translation
Neural Machine Translation of Rare Words with Subword Units

Rico Sennrich and Barry Haddow and Alexandra Birch
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk, bhaddow@inf.ed.ac.uk

ACL 2016
Rare word problem

• Translation is an open-vocabulary task
  • Named entities, numbers, etc.
• Cannot have a fixed pre-defined vocabulary
  • Most MT methods that do so suffer from two issues
    • Out of vocabulary words
    • Rare words
Prior approaches

- Treat all rare words as UNK tokens
- Doesn’t work well for named entities
- Back-off dictionary
  - Replace rare words with UNK during training
  - If system produces UNK, align UNK to a source word and translate (e.g. simply copy)
- Use subword units
Subword units

• Many different ways of construction subword units

• Character n-grams

• Morphological segmentation

• Phoneme or syllable-based segmentation

• Linguistically motivated, but not optimized for task
This paper: use Byte Pair Encodings

• Compression scheme proposed by Gage (1994)

• Start: represent each word as a sequence of characters

• Iteratively merge the most frequent pair of characters into a single symbol

• Provides a balance between vocabulary size and word fragmentation
<table>
<thead>
<tr>
<th>segmentation</th>
<th># tokens</th>
<th># types</th>
<th># UNK</th>
</tr>
</thead>
<tbody>
<tr>
<td>none</td>
<td>100 m</td>
<td>1 750 000</td>
<td>1079</td>
</tr>
<tr>
<td>characters</td>
<td>550 m</td>
<td>3000</td>
<td>0</td>
</tr>
<tr>
<td>character bigrams</td>
<td>306 m</td>
<td>20 000</td>
<td>34</td>
</tr>
<tr>
<td>character trigrams</td>
<td>214 m</td>
<td>120 000</td>
<td>59</td>
</tr>
<tr>
<td>compound splitting△</td>
<td>102 m</td>
<td>1 100 000</td>
<td>643</td>
</tr>
<tr>
<td>morfessor*</td>
<td>109 m</td>
<td>544 000</td>
<td>237</td>
</tr>
<tr>
<td>hyphenation◊</td>
<td>186 m</td>
<td>404 000</td>
<td>230</td>
</tr>
<tr>
<td>BPE</td>
<td>112 m</td>
<td>63 000</td>
<td>0</td>
</tr>
<tr>
<td>BPE (joint)</td>
<td>111 m</td>
<td>82 000</td>
<td>32</td>
</tr>
<tr>
<td>character bigrams</td>
<td>129 m</td>
<td>69 000</td>
<td>34</td>
</tr>
</tbody>
</table>

Table 1: Corpus statistics for German training corpus with different word segmentation techniques. #UNK: number of unknown tokens in newstest2013. △: (Koehn and Knight, 2003); *: (Creutz and Lagus, 2002); ◊: (Liang, 1983).
BPE algorithm

Algorithm 1 Learn BPE operations

```python
import re, collections

def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        for i in range(len(symbols)-1):
            pairs[symbols[i], symbols[i+1]] += freq
    return pairs

def merge_vocab(pair, v_in):
    v_out = {}
    bigram = re.escape(' '.join(pair))
    p = re.compile(r'(?<!\S)' + bigram + r'(?!=\S)')
    for word in v_in:
        w_out = p.sub(''.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out

vocab = ("I o w <\w>": 5, "l o w e r <\w>": 2,
         "n e w e s t <\w>": 6, "w i d e s t <\w>": 3)
num_merges = 10
for i in range(num_merges):
    pairs = get_stats(vocab)
    best = max(pairs, key=pairs.get)
    vocab = merge_vocab(best, vocab)
print(best)
```

(Source: TowardsDataScience)
MT results (En-De)

<table>
<thead>
<tr>
<th>name</th>
<th>segmentation</th>
<th>shortlist</th>
<th>source</th>
<th>vocabulary</th>
<th>target</th>
<th>BLEU single</th>
<th>CHRF3 single</th>
<th>CHRF3 ens-8</th>
<th>unigram $F_1$ (%)</th>
<th>all</th>
<th>rare</th>
<th>OOV</th>
</tr>
</thead>
<tbody>
<tr>
<td>syntax-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Sennrich and Haddow, 2015)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WUnk</td>
<td>-</td>
<td>300 000</td>
<td>500 000</td>
<td>24.4</td>
<td>55.3</td>
<td>20.6</td>
<td>47.2</td>
<td>48.9</td>
<td>56.7</td>
<td>20.4</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>WDict</td>
<td>-</td>
<td>300 000</td>
<td>500 000</td>
<td>22.0</td>
<td>24.2</td>
<td>22.0</td>
<td>50.5</td>
<td>52.4</td>
<td>58.1</td>
<td>36.8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C2-50k</td>
<td>char-bigram</td>
<td>50 000</td>
<td>60 000</td>
<td>22.8</td>
<td>25.3</td>
<td>51.9</td>
<td>53.5</td>
<td>58.4</td>
<td>40.5</td>
<td>30.9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPE-60k</td>
<td>BPE</td>
<td>60 000</td>
<td>60 000</td>
<td>52.0</td>
<td>53.9</td>
<td>58.4</td>
<td>40.9</td>
<td>29.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BPE-J90k</td>
<td>BPE (joint)</td>
<td>90 000</td>
<td>90 000</td>
<td>58.5</td>
<td>41.8</td>
<td>58.5</td>
<td>41.8</td>
<td>33.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: English→German translation performance (BLEU, CHRF3 and unigram $F_1$) on newstest2015. Ens-8: ensemble of 8 models. Best NMT system in bold. Unigram $F_1$ (with ensembles) is computed for all words ($n = 44085$), rare words (not among top 50 000 in training set; $n = 2900$), and OOVs (not in training set; $n = 1168$).
<table>
<thead>
<tr>
<th>Name</th>
<th>Segmentation</th>
<th>Source</th>
<th>Target</th>
<th>BLEU single</th>
<th>BLEU ens-8</th>
<th>CHRF3 single</th>
<th>CHRF3 ens-8</th>
<th>Unigram F₁ (%)</th>
<th>Unigram F₁ (%)</th>
<th>Unigram F₁ (%)</th>
<th>Unigram F₁ (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>phrase-based (Haddow et al., 2015)</td>
<td></td>
<td></td>
<td></td>
<td>24.3</td>
<td>-</td>
<td>53.8</td>
<td>-</td>
<td>56.0</td>
<td>31.3</td>
<td>16.5</td>
<td></td>
</tr>
<tr>
<td>WUnk</td>
<td>-</td>
<td>300 000</td>
<td>500 000</td>
<td>18.8</td>
<td>22.4</td>
<td>46.5</td>
<td>49.9</td>
<td>54.2</td>
<td>25.2</td>
<td>0.0</td>
<td></td>
</tr>
<tr>
<td>WDict</td>
<td>-</td>
<td>300 000</td>
<td>500 000</td>
<td>19.1</td>
<td>22.8</td>
<td>47.5</td>
<td>51.0</td>
<td>54.8</td>
<td>26.5</td>
<td>6.6</td>
<td></td>
</tr>
<tr>
<td>C2-50k</td>
<td>char-bigram</td>
<td>50 000</td>
<td>60 000</td>
<td>20.9</td>
<td>24.1</td>
<td>49.0</td>
<td>51.6</td>
<td>55.2</td>
<td>27.8</td>
<td>17.4</td>
<td></td>
</tr>
<tr>
<td>BPE-60k</td>
<td>BPE</td>
<td>60 000</td>
<td>60 000</td>
<td>20.5</td>
<td>23.6</td>
<td>49.8</td>
<td>52.7</td>
<td>55.3</td>
<td>29.7</td>
<td>15.6</td>
<td></td>
</tr>
<tr>
<td>BPE-J90k</td>
<td>BPE (joint)</td>
<td>90 000</td>
<td>100 000</td>
<td>20.4</td>
<td>24.1</td>
<td>49.7</td>
<td>53.0</td>
<td>55.8</td>
<td>29.7</td>
<td>18.3</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: English→Russian translation performance (BLEU, CHRF3 and unigram F₁) on newstest2015. Ens-8: ensemble of 8 models. Best NMT system in bold. Unigram F₁ (with ensembles) is computed for all words (n = 55654), rare words (not among top 50 000 in training set; n = 5442), and OOVs (not in training set; n = 851).
• BPE helps handle long tail
• Methods like WDict, WUnk fail due to issues like transliteration

Figure 3: English→Russian unigram F$_1$ on newstest2015 plotted by training set frequency rank for different NMT systems.
Beyond MT

- BPE has found use in other tasks too!
- Vaswani et al. (2017) used it with Transformers to fully leverage self-attention
- De-facto representation scheme for large pre-trained language models like GPT, BERT
- Helps alleviate rare word problem
Discussion

• Q1: Based on your reading of the paper, what is the main reason Byte Pair Encoding (BPE) is so effective at handling the rare word problem in MT compared to alternatives like morphological segmentation?

• Q2: List one shortcoming of BPE according to you. How would you try to address/fix it?

• What are some other tasks (not necessarily within NLP) where ideas like sub-word encodings like BPE might be useful?

• Are there other encoding schemes that might work well for producing sub-words?