Handling Rare Words

- Words are a difficult unit to work with: copying can be cumbersome, word vocabularies get very large
- Character-level models don't work well
- Compromise solution: use subword tokens, which may be full words but may also be parts of words

Input: _the **_eco tax** _port i co _in _Po nt - de - Bu is …

Output: _le _port ique **_éco taxe** _de _Pont - de - Bui s

- Can achieve transliteration with this, subword structure makes some translations easier to achieve

Sennrich et al. (2016)

Byte Pair Encoding (BPE)

- Start with every individual byte (character) as its own symbol

```
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
```

- Count bigram character cooccurrences in dictionary
- Merge the most frequent pair of adjacent characters
- Vocabulary stats are weighted over a large corpus
- Doing 30k merges => vocabulary of 30000 word pieces. Includes many whole words:

and there were no re_ fueling stations anywhere

one of the city ’s more un_ princi_ pled real estate agents

Sennrich et al. (2016)

Word Pieces

- Alternative to BPE

while voc size < target voc size:

Build a language model over your corpus

Merge pieces that lead to highest improvement in language model perplexity

- Issues: what LM to use? How to make this tractable?
- SentencePiece library from Google: unigram LM

Schuster and Nakajima (2012), Wu et al. (2016)

Comparison

(a)	**Original:**	furiously
	BPE:	_fur | ious | ly
	Unigram LM:	_fur | ious | ly
(b)	**Original:**	tricycles
	BPE:	_t | ric | y | cles
	Unigram LM:	_tri | cycle | s
(c)	**Original:**	Completely preposterous suggestions
	BPE:	_Comple | t | ely | _prep | ost | erous | _suggest | ions
	Unigram LM:	_Complete | ly | _pre | post | er | ous | _suggestion | s

- BPE produces less linguistically plausible units than word pieces (unigram LM)
- Some evidence that unigram LM works better in pre-trained transformer models
- Other work explores ensembling across multiple tokenizations

Bostrom and Durrett (2020)