* BPE produces less linguistically plausible units than word pieces

* Evidence that unigram LM (WP) works better in pre-trained transformer models

Subword Tokenization

- Handling rare words
  - Words are a difficult unit to work with: copying gets cumbersome, word vocabularies are 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 _ portique _ éco taxe _ de _ Pont - de - Buis ...

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

- Byte-Pair Encoding (BPE)
  - Start w/ 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 coocurrences in dictionary

* Merge the most frequent pair of adjacent characters

  - Vocab stats weighted over a large corpus
  - Doing 30k merges ⟹ vocab of 30,000 word pieces. Includes many whole words.

Ex. and there were no re_fueling stations
one of the city's more un_princi_pled agents

- Word Pieces
  - Alternative to BPE
  - While vocab size < target vocab size, build LM over corpus and merge pieces that lead to highest improvement in LM perplexity