Subword Units Promote Open-vocabulary Translation

Primarily for rare and OOV unseen (English) words

Shuyue Jia
NMT Problem
Fixed vocabulary VS Open vocabulary

1. Out-of-vocabulary (unseen) words, rare words
2. Limited vocabulary, typically 30,000 ~ 50,000
3. Word embedding as a fixed-length vector VS variable-length vector
4. Not always 1-1 correspondence between source and target word

Intuition
Various words are translatable via smaller units e.g., lower → low + er

Traditional Approach
(Word-level NMT model)
Large Vocabulary and Back-off Dictionary

This work
Encode rare / unknown words as sequences of Subword Units
Motivation: “Transparent Translations”

Key: Morphemes (词素) and phonemes (音位) \( \rightarrow \) can translate

Word \( \rightarrow \) Subword Units

1. Named entities
2. Cognates (同源词) and Loanwords (外来词)
3. Morphologically complex words
4. and etc.

Subword Units
Byte Pair (2-gram) Encoding (BPE) Algorithm:

Word → Subword Units

Background

• Philip Gage, 1994
• Data Compression: iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte
• This NMT task: merge characters or character sequences (generate unseen words)

Implications

• Learn compounding and transliteration from subword representations
• Generalize to translate and produce new words (unseen at training time)
Methods

1) Initialize symbol vocabulary with character vocabulary

\[
\text{vocab} = \{\text{low .}: 5, \text{lower .}: 2, \text{new est .}: 6, \text{w i dest .}: 3\}
\]

2) Find the most frequent 2-gram pairs (‘A’, ‘B’) from every word

\[
\{(\text{d', 'e}): 3, (\text{e', 'r}): 2, (\text{l', 'o'): 7, (\text{w', 'r'): 5, (\text{w', 'e'): 8, (\text{e', 'w'): 6,(r', 'i'): 3, (\text{e', 's'): 9, (\text{n', 'o'): 6, (\text{s', 't'): 9,(i', 'd'): 3, (\text{t', 'w'): 7}
\]

We find (‘e’, ‘s’): 9

3) Merge (‘A’, ‘B’) \(\rightarrow\) (‘AB’) and repeat 2).

\[
\{\text{low .}: 5, \text{lower .}: 2, \text{new est .}: 6, \text{w i dest .}: 3\}
\]

4) Stop merging until reach the \text{num(merge operation)} or minimum frequency

Pros

- Balance size(vocabulary) and num(tokens)
Neural Machine Translation of Rare Words with Subword Units

**Model Architecture: Encoder-Decoder**

Input → Subword Units: $x = (x_1, x_2, ......, x_m)$ → Encoder Bi-GRU → Alignment Model → Decoder RNN → Target Sequence → NLL loss → Backprop

Reference Sequence

- **Encoder**
  - Bi-GRU
  - $h_j$
  - $a_{ij}$

- **Decoder**
  - RNN
  - Candidate: $y = (y_1, y_2, ......, y_n)$

Reference: $\hat{y} = (\hat{y}_1, \hat{y}_2, ......, \hat{y}_n)$

**Measurement & Results**
- BLEU (bilingual evaluation understudy)
- ChrF3 (character n-gram F3 score)
- Unigram F1

**BELU**

$$\text{BELU} = \frac{\text{CHR} \cdot \text{CHRP} + \text{CHR}}{\beta} \cdot \text{CHRP}$$

The general formula for the ChrF score is:

$$\text{ChrF} = (1 + \beta^2) \cdot \frac{\text{CHRP} \cdot \text{CHR}}{\beta \cdot \text{CHRP} + \text{CHR}}$$

(1)

where CHRP and CHR stand for character n-gram precision and recall arithmetically averaged over all n-grams:

- **CHRP**
  - percentage of n-grams in the hypothesis which have a counterpart in the reference;

- **CHR**
  - percentage of character n-grams in the reference which are also present in the hypothesis.

and $\beta$ is a parameter which assigns $\beta$ times more importance to recall than to precision – if $\beta = 1$, they have the same importance.

**Conclusions**
- Outperform the back-off dictionary baseline.
- More words to Subwords $\rightarrow$ better performance.
Other Algorithms for “Word → Subword Units”

1. **Byte Pair Encoding**: frequency of the words pair
2. **WordPiece**: probability of size(training set)
3. **Unigram Language Model**
Thanks and have a great weekends!