PHILIPS

www.philips.com

Subword Units Promote Open-vocabulary Translation

Primarily for rare and OOV unseen (English) words

Shuyue Jia

innovation + you

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) → can translate

Word → 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

vocab = {'l o w .': 5, 'l o w e r .': 2, 'n e w e s t .': 6, 'w i d e s t .': 3}

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

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

We find ('e', 's'): 9

3) Merge ('A', 'B') → ('AB') and repeat 2).

Character 2-gram

{' l o w -': 5, 'l o w e r -': 2, ' n e w es t -': 6, ' w i d es t -': 3}

4) Stop merging until reach the *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: $x = (x_1, x_2, \ldots\ldots, x_m)$

Encoder — Bi-GRU

Alignment Model

h_j

α_{ij}

Decoder — RNN

Target Sequence — Candidate: $y = (y_1, y_2, \ldots\ldots, y_n)$

NLL loss

Backprop

Reference Sequence — Reference: $\hat{y} = (\hat{y}_1, \hat{y}_2, \ldots\ldots, \hat{y}_n)$

Measurement & Results

- BLEU (bilingual evaluation understudy)
- ChrF3 (character n-gram F3 score)
- Unigram F1

BELU

Example of poor machine translation output with high precision

Candidate	the	the	the	the	the	the	the
Reference 1	the	cat	is	on	the	mat	
Reference 2	there	is	a	cat	on	the	mat

Of the seven words in the candidate translation, all of ther

$$P = \frac{m}{w_t} = \frac{7}{7} = 1$$

ChrF3

The general formula for the CHRF score is:

$$\text{CHRF}\beta = (1 + \beta^2)\frac{\text{CHRP} \cdot \text{CHRR}}{\beta^2 \cdot \text{CHRP} + \text{CHRR}} \quad (1)$$

where CHRP and CHRR 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;
- CHRR
  percentage of character n-grams in the reference which are also present in the hypothesis.

and β is a parameter which assigns β 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 → 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!

innovation + you