Papers on Neural Translation at subword-levels

Devvrit
Marius

Topics in NLP Fall 2020

Agenda

- Background information
- “Neural Machine Translation of Rare Words with Subword Units” (Sennrich et al.)
- “A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation” (Chung et al.)
- Impact of these papers
- Discussion

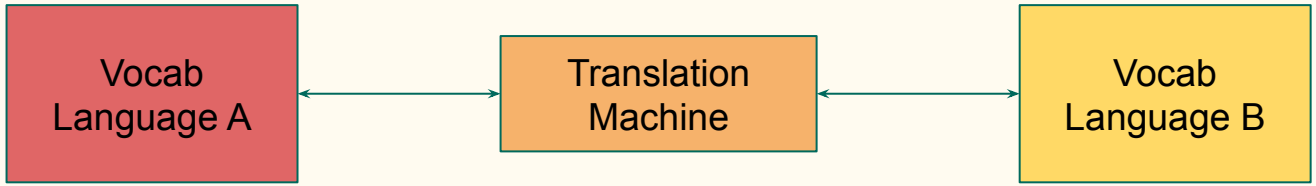
Background information

Vocabulary with 50,000 words covers 95% of text in English (source: Sennrich lecture at University of Edinburgh)

Rare words are high in information

Bad “solution”: Replace Out-Of-Vocabulary with UNK

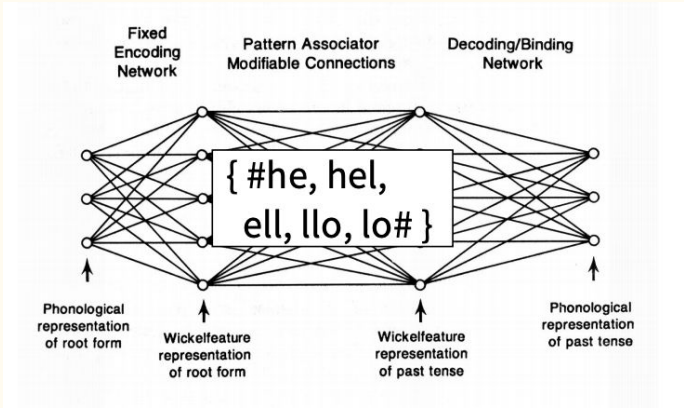
Translation has two sets of vocabulary



History

- An easy alternative is to work with character n-grams
  - Wickelphones (Rumelhart & McClelland 1986)
  - Microsoft's DSSM (Huang, He, Gao, Deng, Acero, & Hect 2013)

- Related idea to use of a convolutional layer



Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Abstract: Sennrich et al, ACL 2016, Neural Machine Translation of Rare Words with Subword Units

- Translation is an open-vocabulary problem.
  - open vocabulary: Not strictly defined by the vocabulary
- Aim is to make the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of subword units
- Consider and compare different word segmentation techniques: n-grams, Byte Pair Encoding (BPE)
- Empirically show that subword models improves over previous works on English →German and English →Russian

Prior work

- Existing works not *solving* the problem

Luong et.al., Addressing the Rare Word Problem in Neural Machine Translation (2015)

- Train an NMT system on data that is augmented by the output of a word alignment algorithm, allowing the NMT system to emit, for each OOV word in the target sentence, the position of its corresponding word in the source sentence.
- This information later utilized in a post-processing step that translates every OOV word using a dictionary.

Prior work

Jean et. al., On Using Very Large Target Vocabulary for Neural Machine Translation (2015)

- Aim is to reduce the training complexity as well as decoding complexity which increase proportionally to the number of target words.

- Propose a method based on importance sampling that allows to use a very large target vocabulary without increasing training complexity.

Ways to form subword units

- n-gram model: take n-length-long sequence of characters and their frequency counts -> Take the top k n-gram(s)
- Byte Pair Encoding:
  - Originally a Compressing Algorithm: Merge frequently occuring bytes together

E.g.,: Compress: “ababcdababcdcdefef...”
a-0, b-1, c-2, d-3, e-4, f-5, “ab” - 6, “cd” - 7 , “ef” - 8, “aba”-9, ...

New word: “13-7-13-...”

Byte Pair Encoding

- A word segmentation algorithm:
  - Though done as bottom up clusering
  - Start with a unigram vocabulary of all (Unicode) characters in data
  - Most frequent ngram pairs ↦ a new ngram

Byte Pair Encoding

- Starts with a vocabulary of characters
- Most frequent ngram pairs -> a new ngram

Dictionary
5 low
2 lower
6 newest
3 widest

Vocabulary:
l, o, w, e, r, n, w, s, t, i, d (Starts with all characters in vocab)

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Byte Pair Encoding

- Starts with a vocabulary of characters
- Most frequent ngram pairs -> a new ngram

Dictionary
5 low
2 lower
6 new**es**t
3 wid**es**t

Vocabulary:
l, o, w, e, r, n, w, s, t, i, d, **es** (Add a pair (e,s) with freq 9)

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Byte Pair Encoding

- Starts with a vocabulary of characters
- Most frequent ngram pairs -> a new ngram

Dictionary
5 low
2 lower
6 new**est**
3 wid**est**

Vocabulary:
l, o, w, e, r, n, w, s, t, i, d, **es**, **est** (Add a pair (es,t) with freq 9)

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Byte Pair Encoding

- Starts with a vocabulary of characters
- Most frequent ngram pairs -> a new ngram

Dictionary
5 **lo**w
2 **lo**wer
6 newest
3 widest

Vocabulary:
l, o, w, e, r, n, w, s, t, i, d, **es**, **est**, **lo** (Add a pair (l,o) with freq 7)

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Rare or unseen words

1) Names
2) Cognates
3) Morphologically complex words

- Compound: book**store**
- Affixation: untouchable

90/100 rare German words were potentially translatable

The vocabulary size tradeoff

Smaller vocab means higher network efficiency

More subwords means longer representation of text

Solutions

- Shortlist and subwords to deal with rare words
- Byte Pair Encoding to learn an efficient vocabulary

Implementation

- C2-50k: Leave lists of k most frequent unsegmented words. Use bi-gram model for rare words
  - k=50,000

- BPE-60k: BPE separately on source and target vocab. Final vocab size = 60k

- BPE-J90k: BPE done on joint vocab. Final vocab size = 90k

Results

name	segmentation	shortlist	vocabulary source	vocabulary target	BLEU single	BLEU ens-8	CHRF3 single	CHRF3 ens-8	unigram F_1 (%) all	unigram F_1 (%) rare	unigram F_1 (%) OOV
syntax-based (Sennrich and Haddow, 2015)					24.4	-	55.3	-	59.1	46.0	37.7
WUnk	-	-	300 000	500 000	20.6	22.8	47.2	48.9	56.7	20.4	0.0
WDict	-	-	300 000	500 000	22.0	24.2	50.5	52.4	58.1	36.8	**36.8**
C2-50k	char-bigram	50 000	60 000	60 000	**22.8**	**25.3**	51.9	53.5	58.4	40.5	30.9
BPE-60k	BPE	-	60 000	60 000	21.5	24.5	**52.0**	53.9	58.4	40.9	29.3
BPE-J90k	BPE (joint)	-	90 000	90 000	**22.8**	24.7	51.7	**54.1**	**58.5**	**41.8**	33.6

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; $n = 1168$).

WDict: word-level model with a back-off dictionary

Wunk: word-level model keeping UNK as UNK

Results

name	segmentation	shortlist	vocabulary source	vocabulary target	BLEU single	BLEU ens-8	CHRF3 single	CHRF3 ens-8	unigram F_1 (%) all	unigram F_1 (%) rare	unigram F_1 (%) OOV
phrase-based (Haddow et al., 2015)					24.3	-	53.8	-	56.0	31.3	16.5
WUnk	-	-	300 000	500 000	18.8	22.4	46.5	49.9	54.2	25.2	0.0
WDict	-	-	300 000	500 000	19.1	22.8	47.5	51.0	54.8	26.5	6.6
C2-50k	char-bigram	50 000	60 000	60 000	**20.9**	**24.1**	49.0	51.6	55.2	27.8	17.4
BPE-60k	BPE	-	60 000	60 000	20.5	23.6	**49.8**	52.7	55.3	29.7	15.6
BPE-J90k	BPE (joint)	-	90 000	100 000	20.4	**24.1**	49.7	**53.0**	**55.8**	**29.7**	**18.3**

Table 3: English→Russian 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 ($n = 55654$), rare words (not among top 50 000 in training set; $n = 5442$), and OOVs (not in training set; $n = 851$).

WDict: word-level model with a back-off dictionary

Wunk: word-level model keeping UNK as UNK

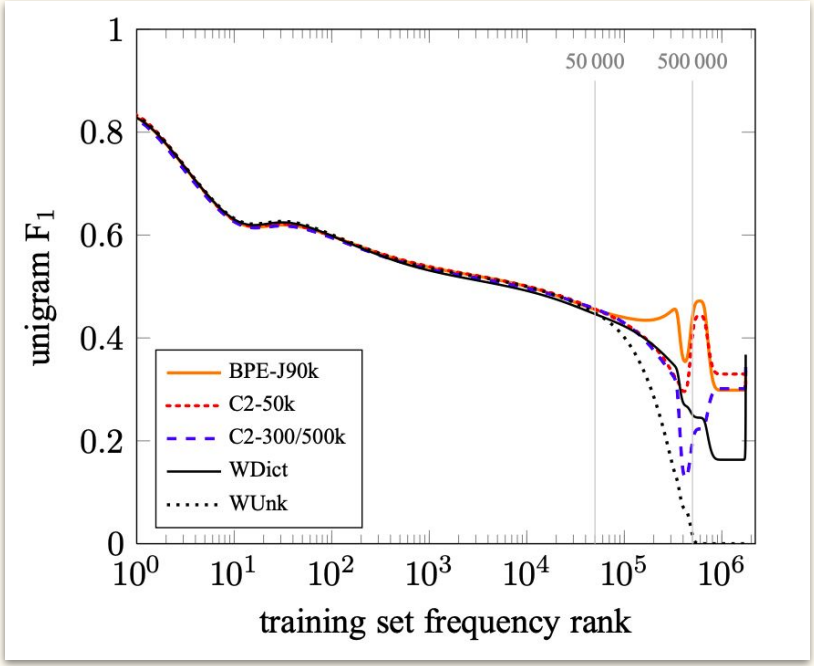
Analysis

BPE-J90k performs well after rank 50,000

Interesting just before rank 500,000

- C2-300/500k drops
- C2-50k remains more stable

Rare words are more sparse than subwords



Translation example

system	sentence
source	health research institutes
reference	Gesundheitsforschungsinstitute
WDict	Forschungsinstitute
C2-50k	Fo|rs|ch|un|gs|in|st|it|ut|io|ne|n
BPE-60k	Gesundheits|forsch|ungsinstitu|ten
BPE-J90k	Gesundheits|forsch|ungsin|stitute

C2-50k shows oversplitting works

BPEs do not match morpheme boundaries

- forschungs|instituten would match morphemes
- forsch|ungsinstitu|ten does not

Conclusion and main contributions

- NMT systems capable of open-vocab translation by using subword units
- Encoding rare words with sub-words is more effective than using large vocabulary
- Using BPE for word segmentation

Abstract: Chung et al, ACL 2016, A Character-Level Decoder without Explicit Segmentation for Neural Machine Translation

- Can NMT generate character sequence without any explicit segmentation?
- Comparing subword-level decoding to character-level decoding
- Forming ensembles achieves SOTA for En-Cs, En-De and En-Fi and perform comparably on En-Ru.

Word-level vs character-level representation

Next few slides

- Reasons for word-level
- Problems with word-level
- Advantages of character-level

Reasons for word-level

Words are units of meaning

Smaller state-space for statistical methods

Vanishing gradient

Problems with word-level

No perfect segmentation algorithm

Vocab needs to include all renditions of the same word

{run, runs, ran, running...}

Cannot achieve distributed representation {run, s, ing...}

Novel words will not be represented

Character-level

Solves the word-level problems

- No segmentation needed
- No vocabulary needed
- Truly open-vocab

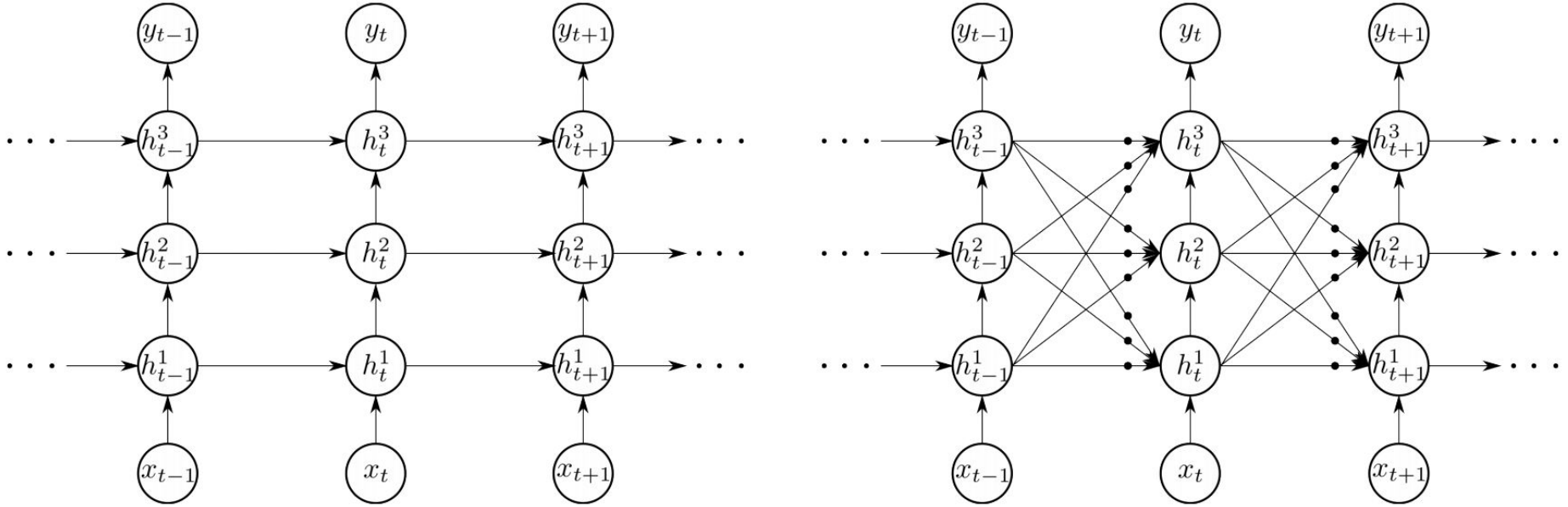
Large state-space not a problem with NMT

Decoder design

1. *Base decoder* - GRU RNN
   - To check if the existing neural network is enough to handle character-level decoding

2. *Bi-scale* RNN (Chung et al. (2015))
   - to see whether it is possible to design a better decoder

Bi-scale RNN (Chung et al. (2015))



(a) Conventional stacked RNN

(b) Gated Feedback RNN

Figure 1. Illustrations of (a) conventional stacking approach and (b) gated-feedback approach to form a deep RNN architecture. Bullets in (b) correspond to global reset gates. Skip connections are omitted to simplify the visualization of networks.

Beam Search

- Works by holding on to many possible branches of outputs before selecting the one with the highest probability

- Size 5 for subword-level; 15 for character-level

Quantitative analysis

English to {German, Czech, Russian, Finnish}

Target	BLEU
BPE	24.83
Char (base)	25.24
Char (bi-scale)	**25.44**

Target	BLEU
BPE	22.96
Char (base)	23.51
Char (bi-scale)	**23.75**

Target	BLEU
BPE	17.61
Char (base)	18.92
Char (bi-scale)	**18.93**

Target	BLEU
BPE	11.73
Char (base)	**13.48**
Char (bi-scale)	13.32

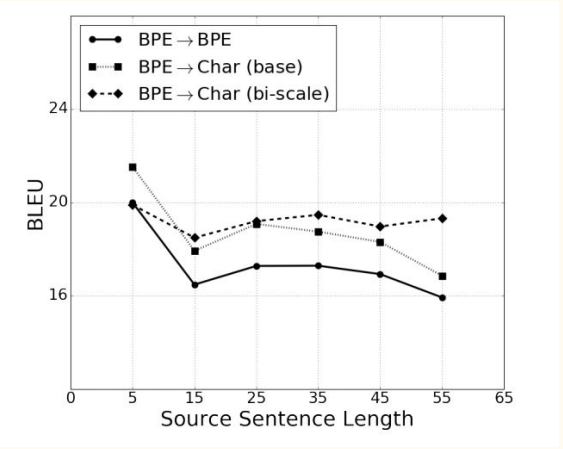
Qualitative analysis

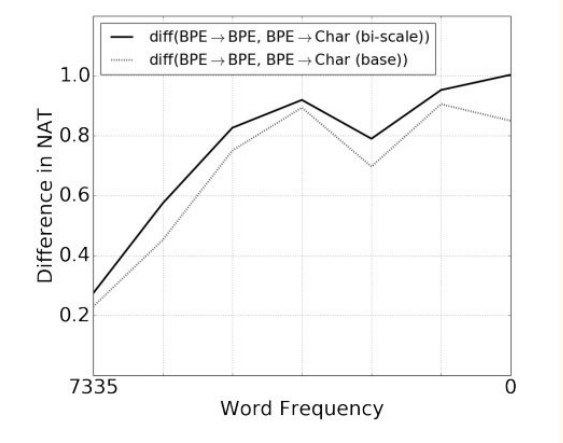
Character-level decoding can form coherent sentences

Character-level decoding helps with rare words

Character-level has slightly slower decoding speed

- Subword 31.9 words/sec
- Char-base 27.5 words/sec
- Char-bi-scale 25.6 words/sec





Conclusion and future work

- Character level decoding works

- Further testing with character-level encoding on encoder side is necessary

How the papers relate to each other

“A character-level decoder..” (Chung) sites “Neural machine translation..” (Senrich)

Same NMT architecture (Bahdanau et al.)

Part of the trend towards smaller tokens

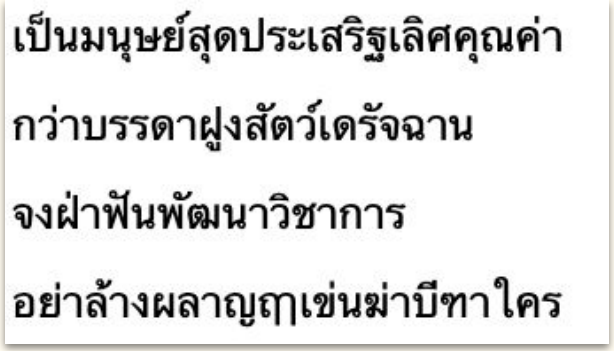
- Words to subwords to characters

Motivation to address rare words

Subword methods on non-spaced languages

Thai

- Words are not separated
- Subword solutions
  - Tokenizer algorithm + BPE
  - Thai Character Cluster (TCC)



Current status

Wordpiece/Sentencepiece model

- Wordpiece models (e,g., BPE) require first tokenizing the words
- Sentencepiece model works from RAW text
  - whitespace are grouped as usual and considered a special token
  - We can reverse things at end by joining pieces and reordering them to spaces

https://github.com/google/sentencepiece

https://arxiv.org/pdf/1804.10959.pdf

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

Current status

BERT, GMT3, etc uses a variant of the wordpiece model

- (Relatively) common words are in dictionary:
  E.g, at, 1910
- Other words are built from wordpieces:
  hypatia = h ##yp ##ati ##a

(Non initial words pieces are represented with two hashes in the start)

Slide from : cs224n (2019 Spring), Prof. Chris Manning, Stanford

How each paper could be improved

“Neural Machine Translation of Rare...” (Sennrich et al.)

- Test more languages

“Character-level decoder...” (Chung et al.)

- Compare to SOTA
- Try character-level at the source

Open for discussion