Neural Machine Translation of Rare World with Subword Units

Rico Sennrich, 2016

Soojung Kim

Contents

- Abstract
- Introduction
- Neural Machine Translation(NMT)
- Subword Translation
- Evaluation
- Analysis
- Conclusion

Abstract

Neural machine translation models operate with a **fixed vocabulary.**

But, translation is an open-vocabulary problem

--> so, Need to word segmentation (n-gram model, BPE)

Neural machine translation (NMT) models typically operate with a fixed vocabulary, but translation is an open-vocabulary problem. Previous work addresses the translation of out-of-vocabulary words by backing off to a dictionary. In this paper, we introduce a simpler and more effective approach, making the NMT model capable of open-vocabulary translation by encoding rare and unknown words as sequences of subword units. This is based on

logical transformations). We discuss the suitability of different word segmentation techniques, including simple character n-gram models and a segmentation based on the *byte pair encoding* compression algorithm, and empirically show that subword models improve over a back-off dictionary baseline for the WMT 15 translation tasks English→German and English→Russian by up to 1.1 and 1.3 BLEU, respectively.

fixed vocabulary : train dataset에 정의한 단어

Introduction

Open-vocabulary problem?

- Out Of Vocabulary (미등록단어, 단어 셋에 없는 단어)
- Ignore Rare word
- Replace OOV words with UNK (Unknown)

translation is open-vocabulary problem

수 많은 단어 유형

- many training corpora contain millions of word types
- productive word formation processes (compounding; derivation) allow formation and understanding of unseen words 복합어, 파생어
- names, numbers are morphologically simple, but open word classes

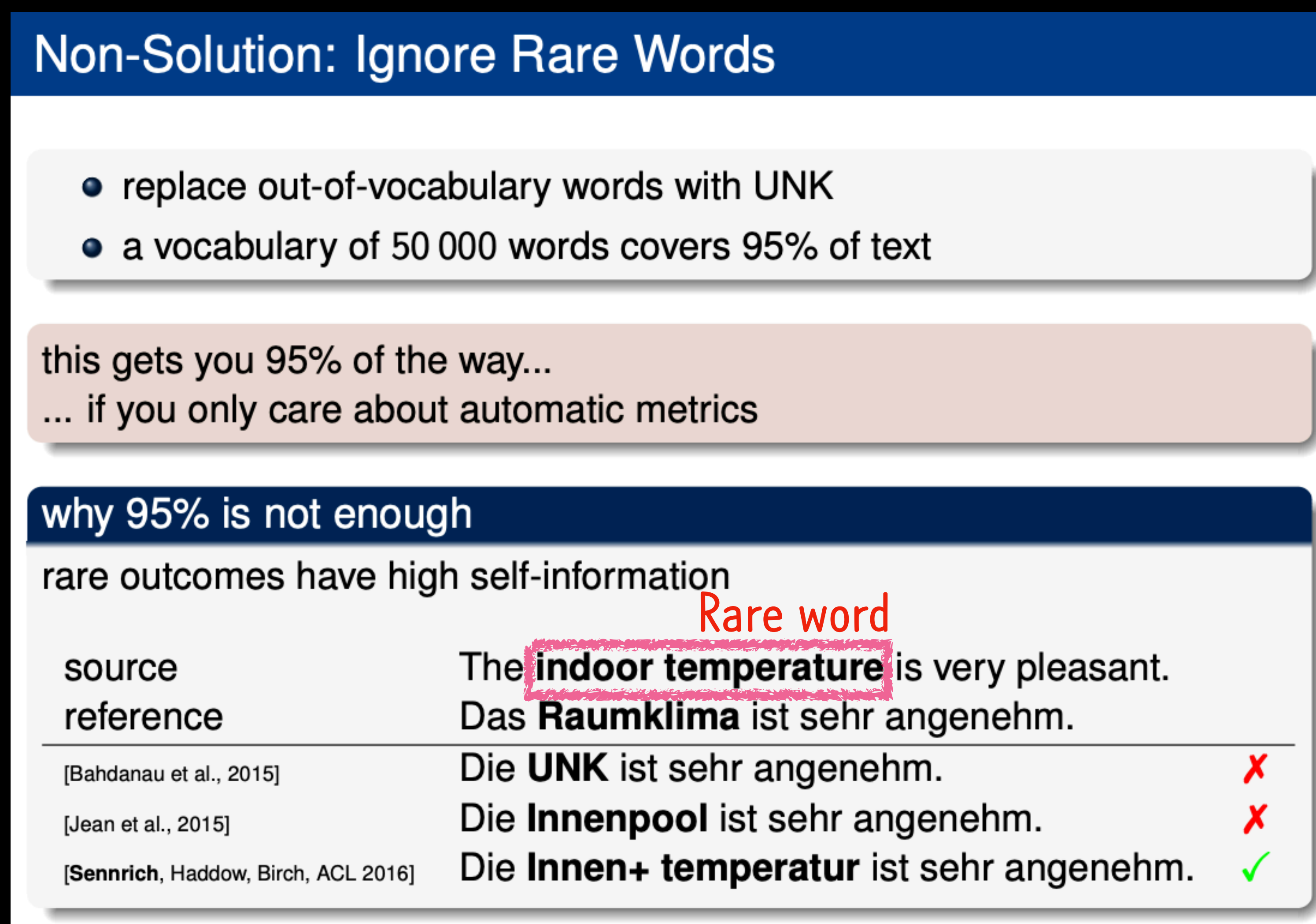
NN이 모르는 단어에 대처하지 못하는 상황

https://ufal.mff.cuni.cz/mtm18/files/02-intro-to-nmt-part2-rico-sennrich.pdf

Introduction

What happens when we ignore Rare Words ?

Rare word를 UNK로 바꾸고 기존 vocabulary로 텍스트의 95% 정도 커버할 수 있지만,
Rare word 자체가 high self-information을 가질 수 있기 때문에 rare word를 무시하면 안된다.



https://ufal.mff.cuni.cz/mtm18/files/02-intro-to-nmt-part2-rico-sennrich.pdf

Introduction

Problem summary

1. Not always a 1-to-1 correspondence between source and target words

2. Word-level models are unable to translate or generate unseen words

https://ufal.mff.cuni.cz/mtm18/files/02-intro-to-nmt-part2-rico-sennrich.pdf

Introduction

Main goal

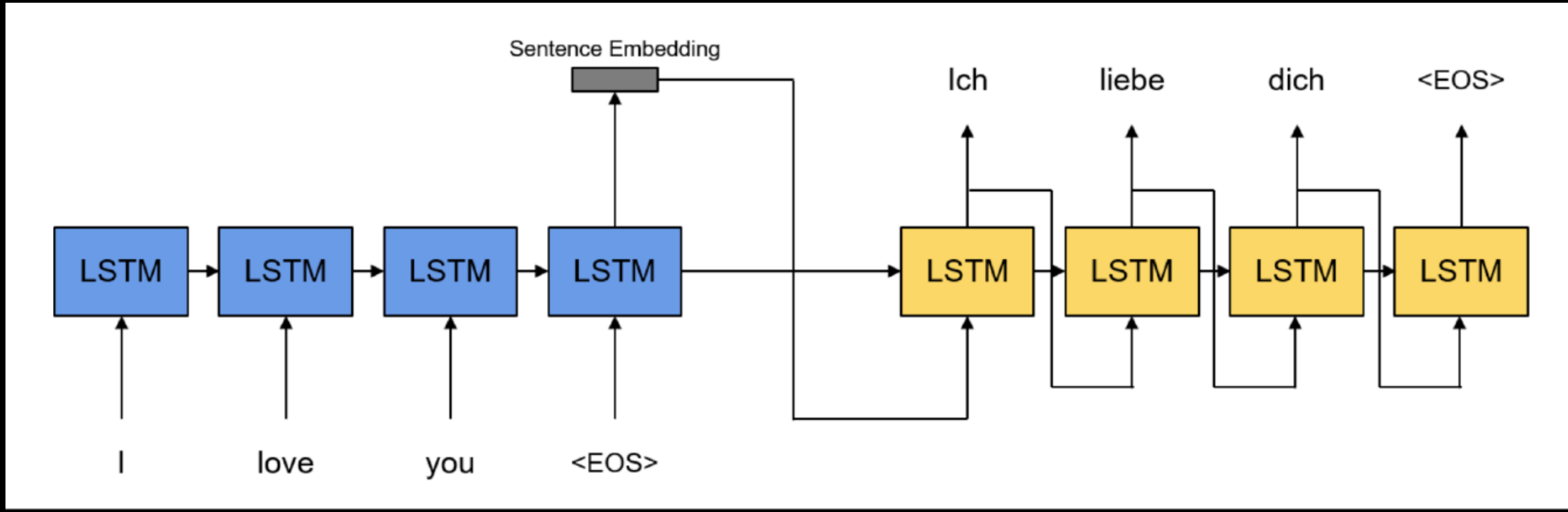
1. Open-vocabulary neural machine translation is possible by encoding(rare) words via subword units

2. BPE allows for the represientation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences
   -> word segmentation strategy

https://ufal.mff.cuni.cz/mtm18/files/02-intro-to-nmt-part2-rico-sennrich.pdf

Neural Machine Translation(NMT)

- Encoder : Bi-directional GRU
  - 번역하고자 하는 소스문장을 특정 임베딩 벡터로 인코딩.
- Decoder : RNN
  - 임베딩된 벡터를 타겟 언어로 번역하여 타겟 문장을 생성.



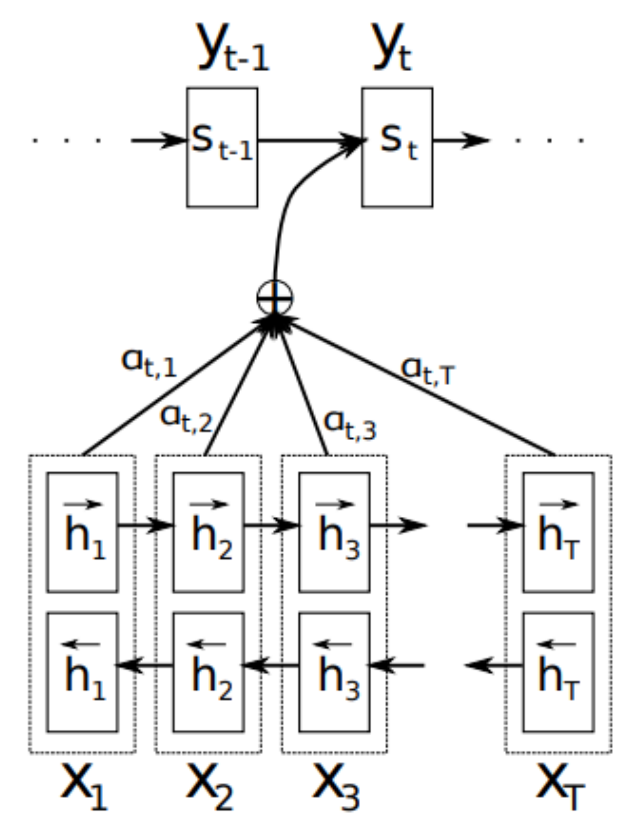


Figure 1: The graphical illustration of the proposed model trying to generate the t-th target word y_t given a source sentence $(x_1, x_2, \ldots, x_T)$.

Neural Machine Translation by jointly learning to align and Translate, Bahdanau

Subword Translation

Segmentation of rare words
into appropriate subword units is sufficient
a.k.a Word piece

Subword Translation

Translation of some words based on translation of known subword units such as morphemes or phoneme

named entities. Between languages that share an alphabet, names can often be copied from source to target text. Transcription or transliteration may be required, especially if the alphabets or syllabaries differ. Example:
Barack Obama (English; German)
Барак Обама (Russian)
バラク・オバマ (ba-ra-ku o-ba-ma) (Japanese)

개체명

cognates and loanwords. Cognates and loanwords with a common origin can differ in regular ways between languages, so that character-level translation rules are sufficient (Tiedemann, 2012). Example:
claustrophobia (English)
Klaustrophobie (German)
Клаустрофобия (Klaustrofobiâ) (Russian)

유사어 및 외래어

morphologically complex words. Words containing multiple morphemes, for instance formed via compounding, affixation, or inflection, may be translatable by translating the morphemes separately. Example:
solar system (English)
Sonnensystem (Sonne + System) (German)
Naprendszer (Nap + Rendszer) (Hungarian)

복합어

Byte Pair Encoding (BPE)

- Data compression technique
- Subword segmentation
  - 단어는 의미를 가진 더 작은 subwords들의 조합으로 이루어진다는 가정

- 어휘 수를 줄일 수 있고, sparsity를 감소시킬 수 있다.

Applying BPE
1. Source
-> compact in text and vocabulary size
-> strong guarantees (seen in the training set)

2. Target vocabulary
-> improves consistency between the source and the target segmentation

Algorithm 1 Learn BPE operations

```
import re, collections

def get_stats(vocab):
  pairs = collections.defaultdict(int)
  for word, freq in vocab.items():
    symbols = word.split()
    for i in range(len(symbols)-1):
      pairs[symbols[i],symbols[i+1]] += freq
  return pairs

def merge_vocab(pair, v_in):
  v_out = {}
  bigram = re.escape(' '.join(pair))
  p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
  for word in v_in:
    w_out = p.sub(''.join(pair), word)
    v_out[w_out] = v_in[word]
  return v_out

vocab = {'l o w </w>' : 5, 'l o w e r </w>' : 2,
         'n e w e s t </w>':6, 'w i d e s t </w>':3}
num_merges = 10
for i in range(num_merges):
  pairs = get_stats(vocab)
  best = max(pairs, key=pairs.get)
  vocab = merge_vocab(best, vocab)
  print(best)
```

r ·	→	r·
l o	→	lo
lo w	→	low
e r·	→	er·

Figure 1: BPE merge operations learned from dictionary {'low', 'lowest', 'newer', 'wider'}.

Byte Pair Encoding (BPE)

```python
import re, collections

def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        for i in range(len(symbols)-1):
            pairs[symbols[i],symbols[i+1]] += freq
    return pairs

def merge_vocab(pair, v_in):
    v_out = {}
    bigram = re.escape(' '.join(pair))
    p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
    for word in v_in:
        w_out = p.sub(''.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out

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

num_merges = 10

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

print(vocab)
```

vocab

1. 맨 뒤에 특수기호 '</w>' 를 넣음.
2. 한 글자(char) 단위로 모두 띄어 초기화.
3. vocab의 value는 빈도수.
   - low는 5번
   - newest는 6번

Byte Pair Encoding (BPE)

```python
import re, collections

def get_stats(vocab):
    pairs = collections.defaultdict(int)
    for word, freq in vocab.items():
        symbols = word.split()
        for i in range(len(symbols)-1):
            pairs[symbols[i],symbols[i+1]] += freq
    return pairs

def merge_vocab(pair, v_in):
    v_out = {}
    bigram = re.escape(' '.join(pair))
    p = re.compile(r'(?<!\S)' + bigram + r'(?!\S)')
    for word in v_in:
        w_out = p.sub(''.join(pair), word)
        v_out[w_out] = v_in[word]
    return v_out

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

num_merges = 10

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

print(vocab)
```

```
iterating 1 / 10 ...defaultdict(<class 'int'>,
          {('d', 'e'): 3,
           ('e', 'r'): 2,
           ('e', 's'): 9,
           ('e', 'w'): 6,
           ('i', 'd'): 3,
           ('l', 'o'): 7,
           ('n', 'e'): 6,
           ('o', 'w'): 7,
           ('r', '_'): 2,
           ('s', 't'): 9,
           ('t', '_'): 9,
           ('w', '_'): 5,
           ('w', 'e'): 8,
           ('w', 'i'): 3})

 best: ('e', 's')
```

```
iterating 2 / 10 ...defaultdict(<class 'int'>,
          {('d', 'es'): 3,
           ('e', 'r'): 2,
           ('e', 'w'): 6,
           ('es', 't'): 9,
           ('i', 'd'): 3,
           ('l', 'o'): 7,
           ('n', 'e'): 6,
           ('o', 'w'): 7,
           ('r', '_'): 2,
           ('t', '_'): 9,
           ('w', '_'): 5,
           ('w', 'e'): 2,
           ('w', 'es'): 6,
           ('w', 'i'): 3})

 best: ('es', 't')
```

best = max(pairs, key=pairs.get)

- 빈도수가 가장 많은 bi-gram을 찾음.
- 찾은 bi-gram을 하나의 unit으로 merge.
- num_merge만큼 반복

Byte Pair Encoding (BPE)

- 토크나이저 입장에서 많이 쓰이는 subwords를 units으로 이용하면 자주 이용되는 단어는 그 자체가 unit이 되며, rare words가 subword units으로 나누어짐.

- BPE는 빈번히 등장하는 substring을 단어로 학습하고, 자주 등장하지 않는 단어들을 최대한 의미보존을 할 수 있는 최소한의 units으로 표현.

```
vocab = {
         'low</w>': 5,
         'low e r </w>': 2,
         'newest</w>': 6,
         'wi d est</w>': 3
        }
```

```
{'low</w>': 5,
 'low': 2,
 'e': 2,
 'r': 2,
 '</w>': 2,
 'newest</w>': 6,
 'wi': 3,
 'd': 3,
 'est</w>': 3}
```

subwords list

Evaluation

1. subword unit으로 representation한 것이 NMT에서 과연 효과가 있는지?
2. vocabulary size, text size, translation quality 측면에서 가장 적합한 word segmentation은?

Experiments

- Datasets
  - WMT 2015 (English, German sentences pair)
  - newsteset2014, 2015 as development set
- Model
  - hidden layer size : 1000
  - embedding layer size : 620
  - shortlist : 30000
  - optimizer : Adadelta
  - mini-batch : 80
  - reshuffle the training-set between epochs

Evaluation

segmentation	# tokens	# types	# UNK
none	100 m	1 750 000	1079
characters	550 m	3000	0
character bigrams	306 m	20 000	34
character trigrams	214 m	120 000	59
compound splitting△	102 m	1 100 000	643
morfessor*	109 m	544 000	237
hyphenation◇	186 m	404 000	230
BPE	112 m	63 000	0
BPE (joint)	111 m	82 000	32
character bigrams (shortlist: 50 000)	129 m	69 000	34

Table 1: Corpus statistics for German training corpus with different word segmentation techniques. #UNK: number of unknown tokens in newstest2013. △: (Koehn and Knight, 2003); *: (Creutz and Lagus, 2002); ◇: (Liang, 1983).

Character n-gram

- trade-offs between sequence length(tokens) and vocabulary size(types)
- way to reduce tokens : most frequent word types unsegmented

BPE

- BPE allows for shorter sequences
- so, attentions model operates on variable-length units

Evaluation

English -> German translation 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 back-off dictionary
- back-off dictionary is incapable of transliterating names

WUnk
- No back-off dictionary, **represents out-of-vocabulary words as UNK**

Subword system (BPE)
- **No back-off dictionary**
- OOV에서 unknown words를 복붙하는 baseline 이 더 좋지만, alphabets이 달라질 때는 subwords가 좋다

BPE-J90k
- learning BPE symbols on **vocabulary union**

BPE-60k
- learning BPE symbols on **separately**

C2-50k
- **Character bigrams**
- shortlist of 50,000 unsegmented words

Analysis

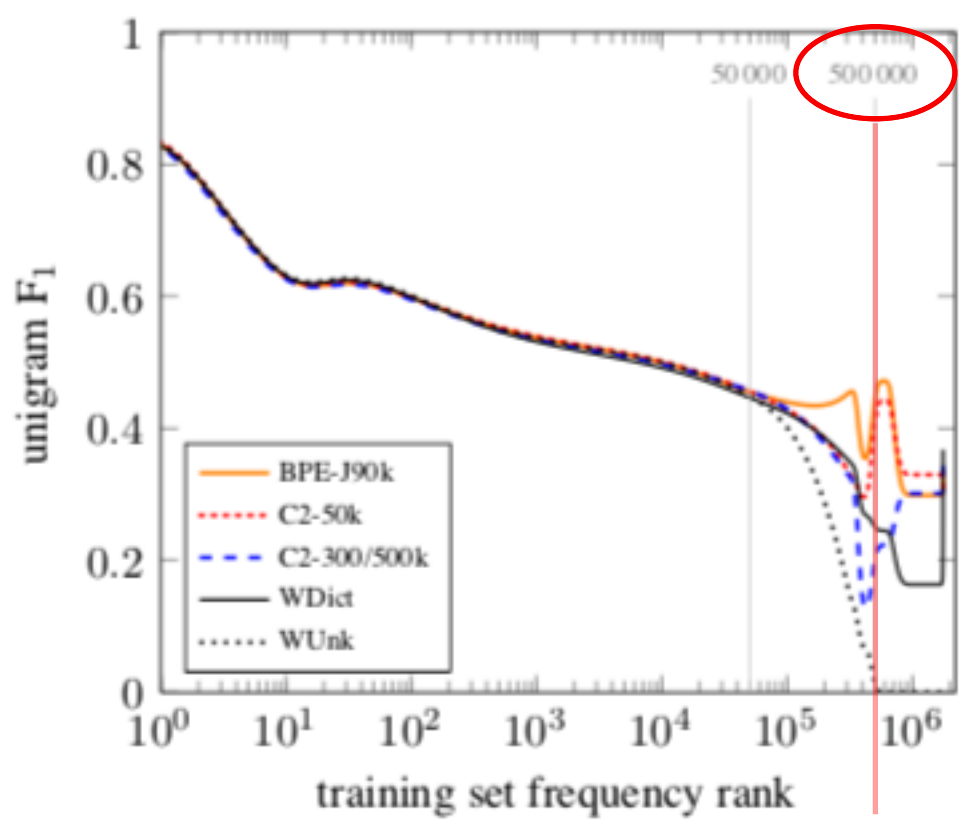


Figure 2: English→German unigram F_1 on newstest2015 plotted by training set frequency rank for different NMT systems.

target vocab : 500,000

- subword system인 C2-3/500k가 back-off dictionary system인 WUnk보다 성능 높음.
- why this result?
  - OOV에서 back-off는 보통 names를 source text에서 copy하는 방식인 반면에, subword는 new words로 바꿔줌.

C2-60k vs C2-3/500k

- only differ in the size of the shortlist

Analysis

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
source	asinine situation
reference	dumme Situation
WDict	asinine situation → UNK → asinine
C2-50k	as|in|in|e situation → As|in|en|si|tu|at|io|n
BPE-60k	as|in|ine situation → A|in|line-|Situation
BPE-J90K	as|in|ine situation → As|in|in-|Situation

Table 4: English→German translation example. "\|" marks subword boundaries.

system	sentence
source	Mirzayeva
reference	Мирзаева (Mirzaeva)
WDict	Mirzayeva → UNK → Mirzayeva
C2-50k	Mi|rz|ay|ev|a → Ми|рз|ае|ва (Mi|rz|ae|va)
BPE-60k	Mirz|ayeva → Мир|за|ева (Mir|za|eva)
BPE-J90k	Mir|za|yeva → Мир|за|ева (Mir|za|eva)
source	rakfisk
reference	ракфиска (rakfiska)
WDict	rakfisk → UNK → rakfisk
C2-50k	ra|kf|is|k → ра|кф|ис|к (ra|kf|is|k)
BPE-60k	rak|f|isk → пра|ф|иск (pra|f|isk)
BPE-J90k	rak|f|isk → рак|ф|иска (rak|f|iska)

Table 5: English→Russian translation examples. "\|" marks subword boundaries.

English -> German

English -> Russian

Conclusion

- BPE는 variable-length의 subword unit으로 word segmentation할 수 있다.
- NMT baseline에서는 OOV, Rare word은 잘 번역되지 않지만 subword model로 Vocab size를 줄이면 성능이 향상된다.
- optimal vocabulary size를 찾는 과정이 필요하다.
- large NMT, back-off models에 의존하지 않는다.