COS 584

Spring 2021

P9: Machine Translation

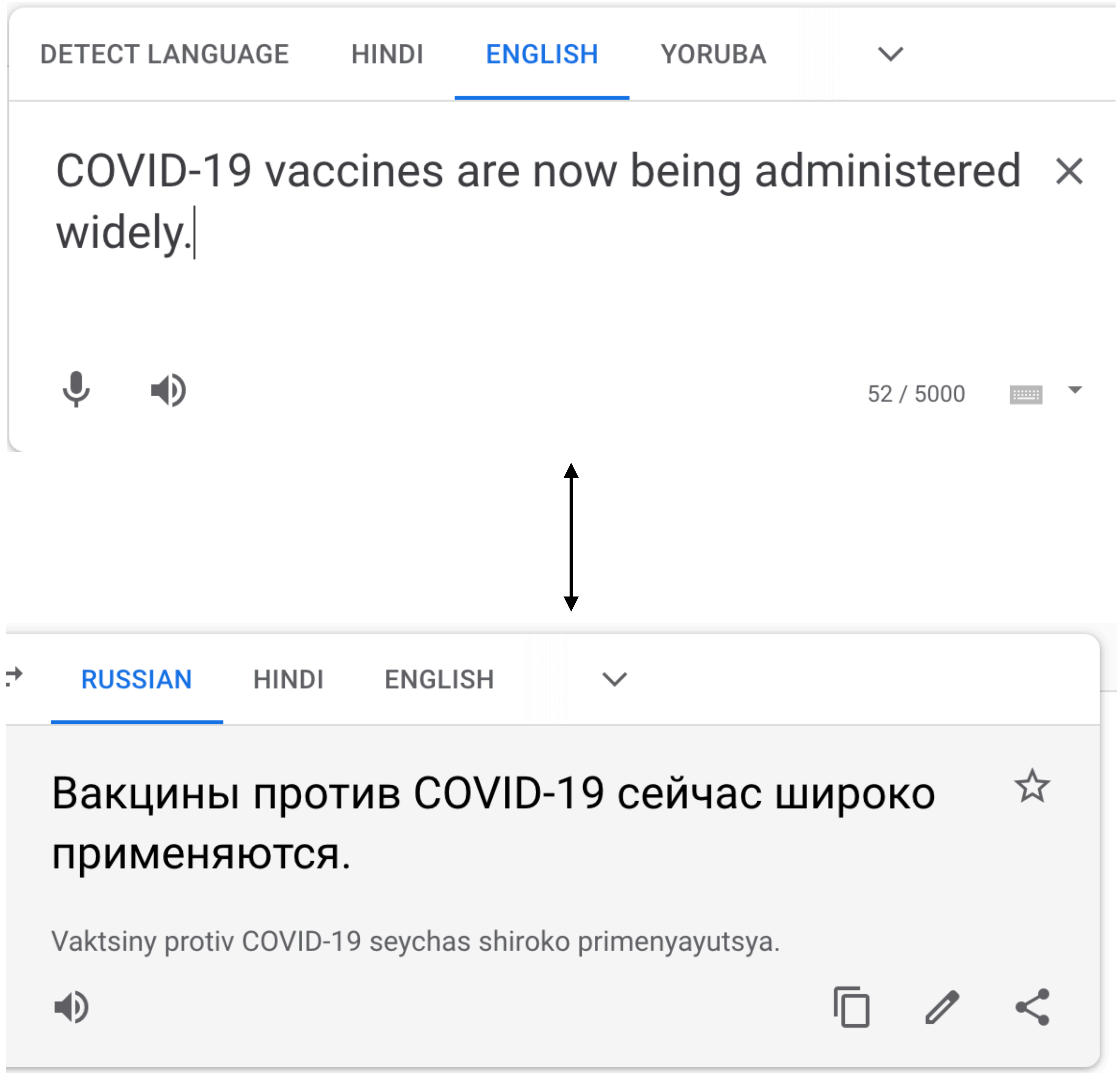
Neural Machine Translation of Rare Words with Subword Units

Rico Sennrich and **Barry Haddow** and **Alexandra Birch**
School of Informatics, University of Edinburgh
{rico.sennrich,a.birch}@ed.ac.uk,bhaddow@inf.ed.ac.uk

ACL 2016

Rare word problem

- Translation is an open-vocabulary task
    - Named entities, numbers, etc.
- Cannot have a fixed pre-defined vocabulary
    - Most MT methods that do so suffer from two issues
        - Out of vocabulary words
        - Rare words



Prior approaches

- Treat all rare words as UNK tokens
  - Doesn't work well for named entities
- Back-off dictionary
  - Replace rare words with UNK during training
  - If system produces UNK, align UNK to a source word and translate (e.g. simply copy)
- Use subword units

Subword units

- Many different ways of construction subword units
  - Character n-grams
  - Morphological segmentation
  - Phoneme or syllable-based segmentation
- Linguistically motivated, but not optimized for task

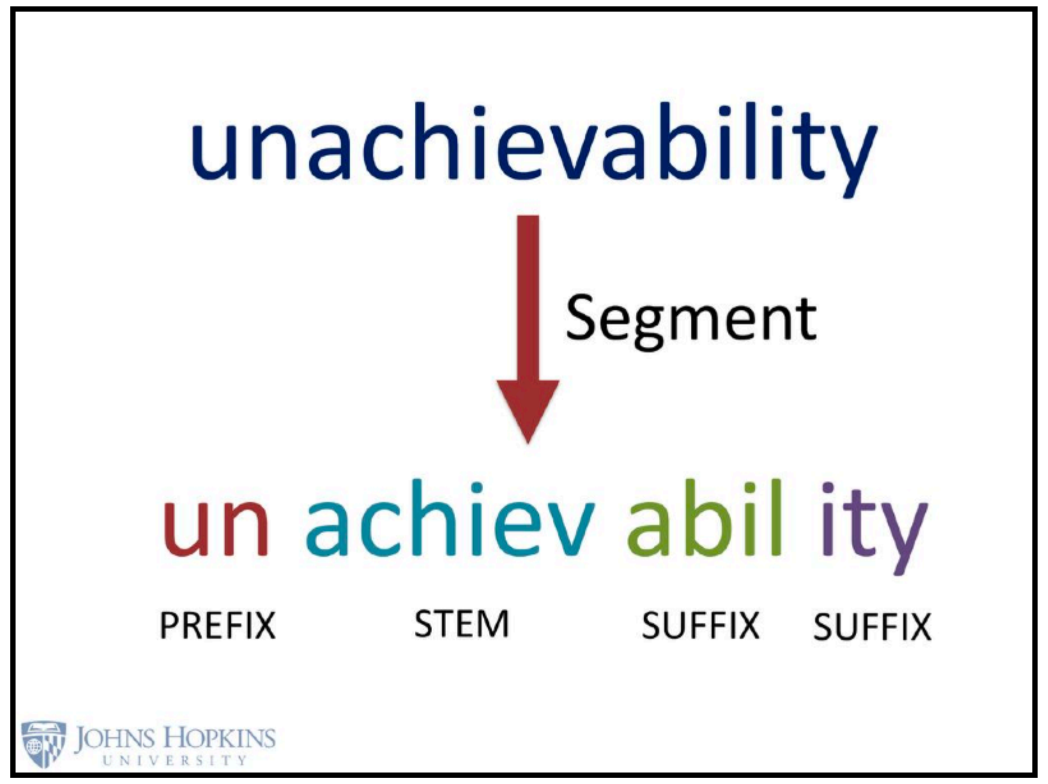
Character N-Grams

- Bigrams for this session's title

Lightweight NLP for Social Media Applications

li	we	tn	or	ia	di	pl	ti
ig	ei	nl	rs	al	ia	li	io
gh	ig	lp	so	lm	aa	ic	on
ht	gh	pf	oc	me	ap	ca	ns
tw	ht	fo	ci	ed	pp	at	

Lithium @btsmith #nlp



THRUSH	[]	T r V S
THRUSHES	[]	T r V S @ z
THRUSHES(2)	[]	T r V S i z
THRUST	[]	T r V s t
THRUSTER	[]	T r V s t 3:r
THRUSTERS	[]	T r V s t 3:r z
THRUSTING	[]	T r V s t i N
THRUSTS	[]	T r V s t s
THRUSTS(2)	[]	T r V s s
THRUSTS(3)	[]	T r V s
THRUWAY	[]	T r u w eI
THS	[]	T s
THUD	[]	T V d
THUG	[]	T V g
THUGGERY	[]	T V g 3:r i:

This paper: use Byte Pair Encodings

- Compression scheme proposed by Gage (1994)
- Start: represent each word as a sequence of characters
- Iteratively merge the most frequent pair of characters into a single symbol
- Provides a balance between vocabulary size and word fragmentation

```
u-n-r-e-l-a-t-e-d
u-n re-l-a-t-e-d
u-n re-l-at-e-d
u-n re-l-at-ed
un re-l-at-ed
un re-l-ated
un rel-ated
un-related
unrelated
```

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).

BPE algorithm

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)
```


(Source: TowardsDataScience)

MT results (En-De)

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; $n = 1168$).

MT results (En-Ru)

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; $n = 851$).

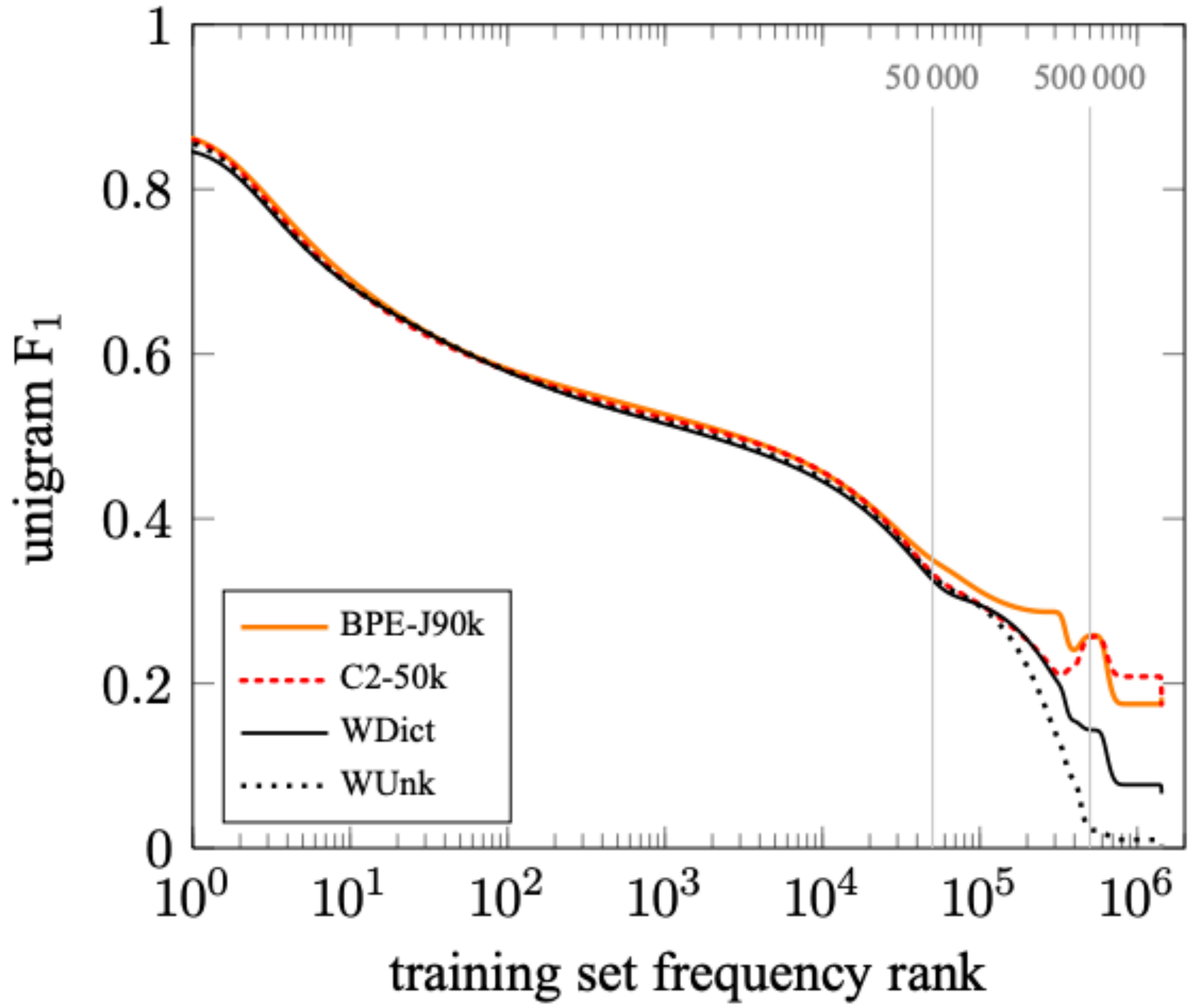


Figure 3: English→Russian unigram F_1 on newstest2015 plotted by training set frequency rank for different NMT systems.

- BPE helps handle long tail
- Methods like WDict, WUnk fail due to issues like transliteration

Beyond MT

Attention Is All You Need

Ashish Vaswani* Google Brain avaswani@google.com
Noam Shazeer* Google Brain noam@google.com
Niki Parmar* Google Research nikip@google.com
Jakob Uszkoreit* Google Research usz@google.com
Llion Jones* Google Research llion@google.com
Aidan N. Gomez* † University of Toronto aidan@cs.toronto.edu
Łukasz Kaiser* Google Brain lukaszkaiser@google.com
Illia Polosukhin* ‡ illia.polosukhin@gmail.com

Improving Language Understanding by Generative Pre-Training

Alec Radford OpenAI alec@openai.com
Karthik Narasimhan OpenAI karthikn@openai.com
Tim Salimans OpenAI tim@openai.com
Ilya Sutskever OpenAI ilyasu@openai.com

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding

Jacob Devlin **Ming-Wei Chang** **Kenton Lee** **Kristina Toutanova**
Google AI Language
{jacobdevlin,mingweichang,kentonl,kristout}@google.com

- BPE has found use in other tasks too!
- Vaswani et al. (2017) used it with Transformers to fully leverage self-attention
- De-facto representation scheme for large pre-trained language models like GPT, BERT
    - Helps alleviate rare word problem

Discussion

- Q1: Based on your reading of the paper, what is the main reason Byte Pair Encoding (BPE) is so effective at handling the rare word problem in MT compared to alternatives like morphological segmentation?
- Q2: List one shortcoming of BPE according to you. How would you try to address/fix it?
- What are some other tasks (not necessarily within NLP) where ideas like sub-word encodings like BPE might be useful?
- Are there other encoding schemes that might work well for producing sub-words?