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

Abstract

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 the intuition that various word classes are translatable via smaller units than words, for instance names (via character copying or transliteration), compounds (via compositional translation), and cognates and loanwords (via phonological and morphological 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.

1 Introduction

Neural machine translation has recently shown impressive results (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). However, the translation of rare words is an open problem. The vocabulary of neural models is typically limited to 30 000–50 000 words, but translation is an open-vocabulary problem, and especially for languages with productive word formation processes such as agglutination and compounding, translation models require mechanisms that go below the word level. As an example, consider compounds such as the German *Abwasser|behandlungs|anlange* 'sewage water treatment plant', for which a segmented, variable-length representation is intuitively more appealing than encoding the word as a fixed-length vector.

For word-level NMT models, the translation of out-of-vocabulary words has been addressed through a back-off to a dictionary look-up (Jean et al., 2015; Luong et al., 2015b). We note that such techniques make assumptions that often do not hold true in practice. For instance, there is not always a 1-to-1 correspondence between source and target words because of variance in the degree of morphological synthesis between languages, like in our introductory compounding example. Also, word-level models are unable to translate or generate unseen words. Copying unknown words into the target text, as done by (Jean et al., 2015; Luong et al., 2015b), is a reasonable strategy for names, but morphological changes and transliteration is often required, especially if alphabets differ.

We investigate NMT models that operate on the level of subword units. Our main goal is to model open-vocabulary translation in the NMT network itself, without requiring a back-off model for rare words. In addition to making the translation process simpler, we also find that the subword models achieve better accuracy for the translation of rare words than large-vocabulary models and back-off dictionaries, and are able to productively generate new words that were not seen at training time. Our analysis shows that the neural networks are able to learn compounding and transliteration from subword representations.

This paper has two main contributions:

- We show that open-vocabulary neural ma-

The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung R&D Institute Poland.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 1715–1725,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics

chine translation is possible by encoding (rare) words via subword units. We find our architecture simpler and more effective than using large vocabularies and back-off dictionaries (Jean et al., 2015; Luong et al., 2015b).

- We adapt *byte pair encoding* (BPE) (Gage, 1994), a compression algorithm, to the task of word segmentation. BPE allows for the representation of an open vocabulary through a fixed-size vocabulary of variable-length character sequences, making it a very suitable word segmentation strategy for neural network models.

2 Neural Machine Translation

We follow the neural machine translation architecture by Bahdanau et al. (2015), which we will briefly summarize here. However, we note that our approach is not specific to this architecture.

The neural machine translation system is implemented as an encoder-decoder network with recurrent neural networks.

The encoder is a bidirectional neural network with gated recurrent units (Cho et al., 2014) that reads an input sequence $x = (x_1, ..., x_m)$ and calculates a forward sequence of hidden states $(\overrightarrow{h}_1, ..., \overrightarrow{h}_m)$, and a backward sequence $(\overleftarrow{h}_1, ..., \overleftarrow{h}_m)$. The hidden states $\overrightarrow{h}_j$ and $\overleftarrow{h}_j$ are concatenated to obtain the annotation vector h_j.

The decoder is a recurrent neural network that predicts a target sequence $y = (y_1, ..., y_n)$. Each word y_i is predicted based on a recurrent hidden state s_i, the previously predicted word y_{i-1}, and a context vector c_i. c_i is computed as a weighted sum of the annotations h_j. The weight of each annotation h_j is computed through an *alignment model* α_{ij}, which models the probability that y_i is aligned to x_j. The alignment model is a single-layer feedforward neural network that is learned jointly with the rest of the network through backpropagation.

A detailed description can be found in (Bahdanau et al., 2015). Training is performed on a parallel corpus with stochastic gradient descent. For translation, a beam search with small beam size is employed.

3 Subword Translation

The main motivation behind this paper is that the translation of some words is transparent in that they are translatable by a competent translator even if they are novel to him or her, based on a translation of known subword units such as morphemes or phonemes. Word categories whose translation is potentially transparent include:

- 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)
- 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)
- 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)

In an analysis of 100 rare tokens (not among the 50 000 most frequent types) in our German training data[1], the majority of tokens are potentially translatable from English through smaller units. We find 56 compounds, 21 names, 6 loanwords with a common origin (*emancipate→emanzipieren*), 5 cases of transparent affixation (*sweetish* 'sweet' + '-ish' → *süßlich* 'süß' + '-lich'), 1 number and 1 computer language identifier.

Our hypothesis is that a segmentation of rare words into appropriate subword units is sufficient to allow for the neural translation network to learn transparent translations, and to generalize this knowledge to translate and produce unseen words.[2] We provide empirical support for this hy-

[1] Primarily parliamentary proceedings and web crawl data.

[2] Not every segmentation we produce is transparent. While we expect no performance benefit from opaque segmentations, i.e. segmentations where the units cannot be translated independently, our NMT models show robustness towards oversplitting.

pothesis in Sections 4 and 5. First, we discuss different subword representations.

3.1 Related Work

For Statistical Machine Translation (SMT), the translation of unknown words has been the subject of intensive research.

A large proportion of unknown words are names, which can just be copied into the target text if both languages share an alphabet. If alphabets differ, transliteration is required (Durrani et al., 2014). Character-based translation has also been investigated with phrase-based models, which proved especially successful for closely related languages (Vilar et al., 2007; Tiedemann, 2009; Neubig et al., 2012).

The segmentation of morphologically complex words such as compounds is widely used for SMT, and various algorithms for morpheme segmentation have been investigated (Nießen and Ney, 2000; Koehn and Knight, 2003; Virpioja et al., 2007; Stallard et al., 2012). Segmentation algorithms commonly used for phrase-based SMT tend to be conservative in their splitting decisions, whereas we aim for an aggressive segmentation that allows for open-vocabulary translation with a compact network vocabulary, and without having to resort to back-off dictionaries.

The best choice of subword units may be task-specific. For speech recognition, phone-level language models have been used (Bazzi and Glass, 2000). Mikolov et al. (2012) investigate subword language models, and propose to use syllables. For multilingual segmentation tasks, multilingual algorithms have been proposed (Snyder and Barzilay, 2008). We find these intriguing, but inapplicable at test time.

Various techniques have been proposed to produce fixed-length continuous word vectors based on characters or morphemes (Luong et al., 2013; Botha and Blunsom, 2014; Ling et al., 2015a; Kim et al., 2015). An effort to apply such techniques to NMT, parallel to ours, has found no significant improvement over word-based approaches (Ling et al., 2015b). One technical difference from our work is that the attention mechanism still operates on the level of words in the model by Ling et al. (2015b), and that the representation of each word is fixed-length. We expect that the attention mechanism benefits from our variable-length representation: the network can learn to place attention on different subword units at each step. Recall our introductory example *Abwasserbehandlungsanlange*, for which a subword segmentation avoids the information bottleneck of a fixed-length representation.

Neural machine translation differs from phrase-based methods in that there are strong incentives to minimize the vocabulary size of neural models to increase time and space efficiency, and to allow for translation without back-off models. At the same time, we also want a compact representation of the text itself, since an increase in text length reduces efficiency and increases the distances over which neural models need to pass information.

A simple method to manipulate the trade-off between vocabulary size and text size is to use shortlists of unsegmented words, using subword units only for rare words. As an alternative, we propose a segmentation algorithm based on byte pair encoding (BPE), which lets us learn a vocabulary that provides a good compression rate of the text.

3.2 Byte Pair Encoding (BPE)

Byte Pair Encoding (BPE) (Gage, 1994) is a simple data compression technique that iteratively replaces the most frequent pair of bytes in a sequence with a single, unused byte. We adapt this algorithm for word segmentation. Instead of merging frequent pairs of bytes, we merge characters or character sequences.

Firstly, we initialize the symbol vocabulary with the character vocabulary, and represent each word as a sequence of characters, plus a special end-of-word symbol '·', which allows us to restore the original tokenization after translation. We iteratively count all symbol pairs and replace each occurrence of the most frequent pair ('A', 'B') with a new symbol 'AB'. Each merge operation produces a new symbol which represents a character n-gram. Frequent character n-grams (or whole words) are eventually merged into a single symbol, thus BPE requires no shortlist. The final symbol vocabulary size is equal to the size of the initial vocabulary, plus the number of merge operations – the latter is the only hyperparameter of the algorithm.

For efficiency, we do not consider pairs that cross word boundaries. The algorithm can thus be run on the dictionary extracted from a text, with each word being weighted by its frequency. A minimal Python implementation is shown in Al-

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'}.

gorithm 1. In practice, we increase efficiency by indexing all pairs, and updating data structures incrementally.

The main difference to other compression algorithms, such as Huffman encoding, which have been proposed to produce a variable-length encoding of words for NMT (Chitnis and DeNero, 2015), is that our symbol sequences are still interpretable as subword units, and that the network can generalize to translate and produce new words (unseen at training time) on the basis of these subword units.

Figure 1 shows a toy example of learned BPE operations. At test time, we first split words into sequences of characters, then apply the learned operations to merge the characters into larger, known symbols. This is applicable to any word, and allows for open-vocabulary networks with fixed symbol vocabularies.[3] In our example, the OOV 'lower' would be segmented into 'low er·'.

[3]The only symbols that will be unknown at test time are unknown characters, or symbols of which all occurrences in the training text have been merged into larger symbols, like 'safeguar', which has all occurrences in our training text merged into 'safeguard'. We observed no such symbols at test time, but the issue could be easily solved by recursively reversing specific merges until all symbols are known.

We evaluate two methods of applying BPE: learning two independent encodings, one for the source, one for the target vocabulary, or learning the encoding on the union of the two vocabularies (which we call *joint BPE*).[4] The former has the advantage of being more compact in terms of text and vocabulary size, and having stronger guarantees that each subword unit has been seen in the training text of the respective language, whereas the latter improves consistency between the source and the target segmentation. If we apply BPE independently, the same name may be segmented differently in the two languages, which makes it harder for the neural models to learn a mapping between the subword units. To increase the consistency between English and Russian segmentation despite the differing alphabets, we transliterate the Russian vocabulary into Latin characters with ISO-9 to learn the joint BPE encoding, then transliterate the BPE merge operations back into Cyrillic to apply them to the Russian training text.[5]

4 Evaluation

We aim to answer the following empirical questions:

- Can we improve the translation of rare and unseen words in neural machine translation by representing them via subword units?
- Which segmentation into subword units performs best in terms of vocabulary size, text size, and translation quality?

We perform experiments on data from the shared translation task of WMT 2015. For English→German, our training set consists of 4.2 million sentence pairs, or approximately 100 million tokens. For English→Russian, the training set consists of 2.6 million sentence pairs, or approximately 50 million tokens. We tokenize and truecase the data with the scripts provided in Moses (Koehn et al., 2007). We use newstest2013 as development set, and report results on newstest2014 and newstest2015.

We report results with BLEU (*mteval-v13a.pl*), and CHRF3 (Popović, 2015), a character n-gram F_3 score which was found to correlate well with

[4]In practice, we simply concatenate the source and target side of the training set to learn joint BPE.

[5]Since the Russian training text also contains words that use the Latin alphabet, we also apply the Latin BPE operations.

human judgments, especially for translations out of English (Stanojević et al., 2015). Since our main claim is concerned with the translation of rare and unseen words, we report separate statistics for these. We measure these through unigram F_1, which we calculate as the harmonic mean of clipped unigram precision and recall.[6]

We perform all experiments with Groundhog[7] (Bahdanau et al., 2015). We generally follow settings by previous work (Bahdanau et al., 2015; Jean et al., 2015). All networks have a hidden layer size of 1000, and an embedding layer size of 620. Following Jean et al. (2015), we only keep a shortlist of $\tau = 30000$ words in memory.

During training, we use Adadelta (Zeiler, 2012), a minibatch size of 80, and reshuffle the training set between epochs. We train a network for approximately 7 days, then take the last 4 saved models (models being saved every 12 hours), and continue training each with a fixed embedding layer (as suggested by (Jean et al., 2015)) for 12 hours. We perform two independent training runs for each models, once with cut-off for gradient clipping (Pascanu et al., 2013) of 5.0, once with a cut-off of 1.0 – the latter produced better single models for most settings. We report results of the system that performed best on our development set (newstest2013), and of an ensemble of all 8 models.

We use a beam size of 12 for beam search, with probabilities normalized by sentence length. We use a bilingual dictionary based on fast-align (Dyer et al., 2013). For our baseline, this serves as back-off dictionary for rare words. We also use the dictionary to speed up translation for all experiments, only performing the softmax over a filtered list of candidate translations (like Jean et al. (2015), we use $K = 30000$; $K' = 10$).

4.1 Subword statistics

Apart from translation quality, which we will verify empirically, our main objective is to represent an open vocabulary through a compact fixed-size subword vocabulary, and allow for efficient training and decoding.[8]

Statistics for different segmentations of the German side of the parallel data are shown in Table 1. A simple baseline is the segmentation of words into character n-grams.[9] Character n-grams allow for different trade-offs between sequence length (# tokens) and vocabulary size (# types), depending on the choice of n. The increase in sequence length is substantial; one way to reduce sequence length is to leave a shortlist of the k most frequent word types unsegmented. Only the unigram representation is truly open-vocabulary. However, the unigram representation performed poorly in preliminary experiments, and we report translation results with a bigram representation, which is empirically better, but unable to produce some tokens in the test set with the training set vocabulary.

We report statistics for several word segmentation techniques that have proven useful in previous SMT research, including frequency-based compound splitting (Koehn and Knight, 2003), rule-based hyphenation (Liang, 1983), and Morfessor (Creutz and Lagus, 2002). We find that they only moderately reduce vocabulary size, and do not solve the unknown word problem, and we thus find them unsuitable for our goal of open-vocabulary translation without back-off dictionary.

BPE meets our goal of being open-vocabulary, and the learned merge operations can be applied to the test set to obtain a segmentation with no unknown symbols.[10] Its main difference from the character-level model is that the more compact representation of BPE allows for shorter sequences, and that the attention model operates on variable-length units.[11] Table 1 shows BPE with 59 500 merge operations, and joint BPE with 89 500 operations.

In practice, we did not include infrequent subword units in the NMT network vocabulary, since there is noise in the subword symbol sets, e.g. because of characters from foreign alphabets. Hence, our network vocabularies in Table 2 are typically slightly smaller than the number of types in Table 1.

[6] Clipped unigram precision is essentially 1-gram BLEU without brevity penalty.

[7] `github.com/sebastien-j/LV_groundhog`

[8] The time complexity of encoder-decoder architectures is at least linear to sequence length, and oversplitting harms efficiency.

[9] Our character n-grams do not cross word boundaries. We mark whether a subword is word-final or not with a special character, which allows us to restore the original tokenization.

[10] Joint BPE can produce segments that are unknown because they only occur in the English training text, but these are rare (0.05% of test tokens).

[11] We highlighted the limitations of word-level attention in section 3.1. At the other end of the spectrum, the character level is suboptimal for alignment (Tiedemann, 2009).

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; $n = 1168$).

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$^{\triangle}$	102 m	1 100 000	643
morfessor*	109 m	544 000	237
hyphenation$^{\diamond}$	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. $\triangle$: (Koehn and Knight, 2003); *: (Creutz and Lagus, 2002); $\diamond$: (Liang, 1983).

4.2 Translation experiments

English→German translation results are shown in Table 2; English→Russian results in Table 3.

Our baseline **WDict** is a word-level model with a back-off dictionary. It differs from **WUnk** in that the latter uses no back-off dictionary, and just represents out-of-vocabulary words as UNK[12]. The back-off dictionary improves unigram F_1 for rare and unseen words, although the improvement is smaller for English→Russian, since the back-off dictionary is incapable of transliterating names.

All subword systems operate without a back-off dictionary. We first focus on unigram F_1, where all systems improve over the baseline, especially for rare words (36.8%→41.8% for EN→DE; 26.5%→29.7% for EN→RU). For OOVs, the baseline strategy of copying unknown words works well for English→German. However, when alphabets differ, like in English→Russian, the subword models do much better.

[12]We use *UNK* for words that are outside the model vocabulary, and *OOV* for those that do not occur in the training text.

Unigram F_1 scores indicate that learning the BPE symbols on the vocabulary union (**BPE-J90k**) is more effective than learning them separately (**BPE-60k**), and more effective than using character bigrams with a shortlist of 50 000 unsegmented words (**C2-50k**), but all reported subword segmentations are viable choices and outperform the back-off dictionary baseline.

Our subword representations cause big improvements in the translation of rare and unseen words, but these only constitute 9-11% of the test sets. Since rare words tend to carry central information in a sentence, we suspect that BLEU and CHRF3 underestimate their effect on translation quality. Still, we also see improvements over the baseline in total unigram F_1, as well as BLEU and CHRF3, and the subword ensembles outperform the WDict baseline by 0.3–1.3 BLEU and 0.6–2 CHRF3. There is some inconsistency between BLEU and CHRF3, which we attribute to the fact that BLEU has a precision bias, and CHRF3 a recall bias.

For English→German, we observe the best BLEU score of 25.3 with C2-50k, but the best CHRF3 score of 54.1 with BPE-J90k. For comparison to the (to our knowledge) best non-neural MT system on this data set, we report syntax-based SMT results (Sennrich and Haddow, 2015). We observe that our best systems outperform the syntax-based system in terms of BLEU, but not in terms of CHRF3. Regarding other neural systems, Luong et al. (2015a) report a BLEU score of 25.9 on newstest2015, but we note that they use an ensemble of 8 independently trained models, and also report strong improvements from applying dropout, which we did not use. We are confident that our improvements to the translation of rare words are orthogonal to improvements achievable through other improvements in the network archi-

tecture, training algorithm, or better ensembles.

For English→Russian, the state of the art is the phrase-based system by Haddow et al. (2015). It outperforms our WDict baseline by 1.5 BLEU. The subword models are a step towards closing this gap, and BPE-J90k yields an improvement of 1.3 BLEU, and 2.0 CHRF3, over WDict.

As a further comment on our translation results, we want to emphasize that performance variability is still an open problem with NMT. On our development set, we observe differences of up to 1 BLEU between different models. For single systems, we report the results of the model that performs best on dev (out of 8), which has a stabilizing effect, but how to control for randomness deserves further attention in future research.

5 Analysis

5.1 Unigram accuracy

Our main claims are that the translation of rare and unknown words is poor in word-level NMT models, and that subword models improve the translation of these word types. To further illustrate the effect of different subword segmentations on the translation of rare and unseen words, we plot target-side words sorted by their frequency in the training set.[13] To analyze the effect of vocabulary size, we also include the system **C2-3/500k**, which is a system with the same vocabulary size as the WDict baseline, and character bigrams to represent unseen words.

Figure 2 shows results for the English–German ensemble systems on newstest2015. Unigram F_1 of all systems tends to decrease for lower-frequency words. The baseline system has a spike in F_1 for OOVs, i.e. words that do not occur in the training text. This is because a high proportion of OOVs are names, for which a copy from the source to the target text is a good strategy for English→German.

The systems with a target vocabulary of 500 000 words mostly differ in how well they translate words with rank > 500 000. A back-off dictionary is an obvious improvement over producing UNK, but the subword system C2-3/500k achieves better performance. Note that all OOVs that the back-off dictionary produces are words that are copied from the source, usually names, while the subword systems can productively form new words such as compounds.

For the 50 000 most frequent words, the representation is the same for all neural networks, and all neural networks achieve comparable unigram F_1 for this category. For the interval between frequency rank 50 000 and 500 000, the comparison between C2-3/500k and C2-50k unveils an interesting difference. The two systems only differ in the size of the shortlist, with C2-3/500k representing words in this interval as single units, and C2-50k via subword units. We find that the performance of C2-3/500k degrades heavily up to frequency rank 500 000, at which point the model switches to a subword representation and performance recovers. The performance of C2-50k remains more stable. We attribute this to the fact that subword units are less sparse than words. In our training set, the frequency rank 50 000 corresponds to a frequency of 60 in the training data; the frequency rank 500 000 to a frequency of 2. Because subword representations are less sparse, reducing the size of the network vocabulary, and representing more words via subword units, can lead to better performance.

The F_1 numbers hide some qualitative differences between systems. For English→German, WDict produces few OOVs (26.5% recall), but with high precision (60.6%) , whereas the subword systems achieve higher recall, but lower precision. We note that the character bigram model C2-50k produces the most OOV words, and achieves relatively low precision of 29.1% for this category. However, it outperforms the back-off dictionary in recall (33.0%). BPE-60k, which suffers from transliteration (or copy) errors due to segmentation inconsistencies, obtains a slightly better precision (32.4%), but a worse recall (26.6%). In contrast to BPE-60k, the joint BPE encoding of BPE-J90k improves both precision (38.6%) and recall (29.8%).

For English→Russian, unknown names can only rarely be copied, and usually require transliteration. Consequently, the WDict baseline performs more poorly for OOVs (9.2% precision; 5.2% recall), and the subword models improve both precision and recall (21.9% precision and 15.6% recall for BPE-J90k). The full unigram F_1 plot is shown in Figure 3.

[13] We perform binning of words with the same training set frequency, and apply bezier smoothing to the graph.

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; $n = 851$).

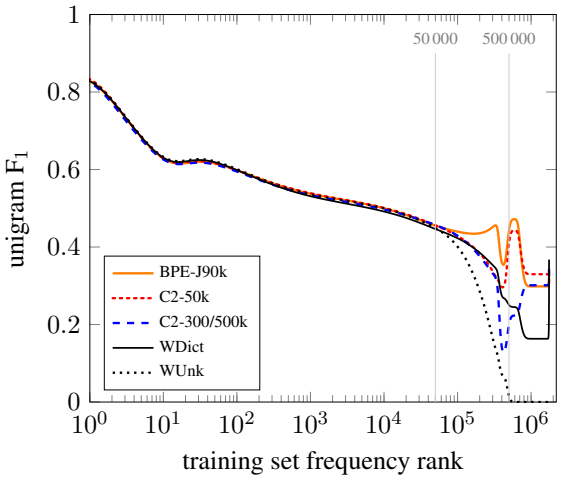

Figure 2: English→German unigram F_1 on newstest2015 plotted by training set frequency rank for different NMT systems.

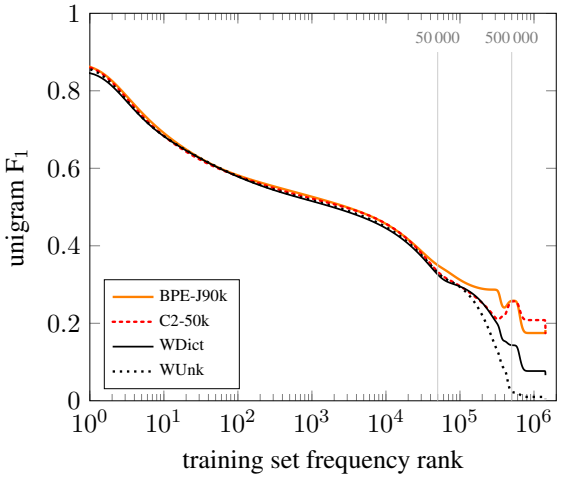

Figure 3: English→Russian unigram F_1 on newstest2015 plotted by training set frequency rank for different NMT systems.

5.2 Manual Analysis

Table 4 shows two translation examples for the translation direction English→German, Table 5 for English→Russian. The baseline system fails for all of the examples, either by deleting content (*health*), or by copying source words that should be translated or transliterated. The subword translations of *health research institutes* show that the subword systems are capable of learning translations when oversplitting (*research→Fo|rs|ch|un|g*), or when the segmentation does not match morpheme boundaries: the segmentation *Forschungs|instituten* would be linguistically more plausible, and simpler to align to the English *research institutes*, than the segmentation *Forsch|ungsinstitu|ten* in the BPE-60k system, but still, a correct translation is produced. If the systems have failed to learn a translation due to data sparseness, like for *asinine*, which should be translated as *dumm*, we see translations that are wrong, but could be plausible for (partial) loanwords (*asinine Situation→Asinin-Situation*).

The English→Russian examples show that the subword systems are capable of transliteration. However, transliteration errors do occur, either due to ambiguous transliterations, or because of non-consistent segmentations between source and target text which make it hard for the system to learn a transliteration mapping. Note that the BPE-60k system encodes *Mirzayeva* inconsistently for the two language pairs (*Mirz|ayeva*→Мир|за|ева *Mir|za|eva*). This example is still translated correctly, but we observe spurious insertions and deletions of characters in the BPE-60k system. An example is the transliteration of *rakfisk*, where a п is inserted and a к is deleted. We trace this error back to translation pairs in the training data with inconsistent segmentations, such as (*p|rak|ri|ti*→пра|крит|и

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.

(pra|krit|i)), from which the translation (*rak*→пра) is erroneously learned. The segmentation of the joint BPE system (BPE-J90k) is more consistent (*pra*|*krit*|*i*→пра|крит|и (pra|krit|i)).

6 Conclusion

The main contribution of this paper is that we show that neural machine translation systems are capable of open-vocabulary translation by representing rare and unseen words as a sequence of subword units.[14] This is both simpler and more effective than using a back-off translation model. We introduce a variant of byte pair encoding for word segmentation, which is capable of encoding open vocabularies with a compact symbol vocabulary of variable-length subword units. We show performance gains over the baseline with both BPE segmentation, and a simple character bigram segmentation.

Our analysis shows that not only out-of-vocabulary words, but also rare in-vocabulary words are translated poorly by our baseline NMT system, and that reducing the vocabulary size of subword models can actually improve performance. In this work, our choice of vocabulary size is somewhat arbitrary, and mainly motivated by comparison to prior work. One avenue of future research is to learn the optimal vocabulary size for a translation task, which we expect to depend on the language pair and amount of training data, automatically. We also believe there is further potential in bilingually informed segmentation algorithms to create more alignable subword units, although the segmentation algorithm cannot rely on the target text at runtime.

While the relative effectiveness will depend on language-specific factors such as vocabulary size, we believe that subword segmentations are suitable for most language pairs, eliminating the need for large NMT vocabularies or back-off models.

[14]The source code of the segmentation algorithms is available at `https://github.com/rsennrich/subword-nmt`.

Acknowledgments

We thank Maja Popović for her implementation of CHRF, with which we verified our re-implementation. The research presented in this publication was conducted in cooperation with Samsung Electronics Polska sp. z o.o. - Samsung R&D Institute Poland. This project received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement 645452 (QT21).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Issam Bazzi and James R. Glass. 2000. Modeling out-of-vocabulary words for robust speech recognition. In *Sixth International Conference on Spoken Language Processing, ICSLP 2000 / INTERSPEECH 2000*, pages 401–404, Beijing, China.

Jan A. Botha and Phil Blunsom. 2014. Compositional Morphology for Word Representations and Language Modelling. In *Proceedings of the 31st International Conference on Machine Learning (ICML)*, Beijing, China.

Rohan Chitnis and John DeNero. 2015. Variable-Length Word Encodings for Neural Translation Models. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Mathias Creutz and Krista Lagus. 2002. Unsupervised Discovery of Morphemes. In *Proceedings of the ACL-02 Workshop on Morphological and Phonological Learning*, pages 21–30. Association for Computational Linguistics.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, pages 148–153, Gothenburg, Sweden.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia. Association for Computational Linguistics.

Philip Gage. 1994. A New Algorithm for Data Compression. *C Users J.*, 12(2):23–38, February.

Barry Haddow, Matthias Huck, Alexandra Birch, Nikolay Bogoychev, and Philipp Koehn. 2015. The Edinburgh/JHU Phrase-based Machine Translation Systems for WMT 2015. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 126–133, Lisbon, Portugal. Association for Computational Linguistics.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On Using Very Large Target Vocabulary for Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1–10, Beijing, China. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent Continuous Translation Models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle. Association for Computational Linguistics.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. 2015. Character-Aware Neural Language Models. *CoRR*, abs/1508.06615.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *EACL '03: Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics*, pages 187–193, Budapest, Hungary. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the ACL-2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Franklin M. Liang. 1983. *Word hy-phen-a-tion by com-put-er*. Ph.D. thesis, Stanford University, Department of Linguistics, Stanford, CA.

Wang Ling, Chris Dyer, Alan W. Black, Isabel Trancoso, Ramon Fermandez, Silvio Amir, Luis Marujo, and Tiago Luis. 2015a. Finding Function in Form: Compositional Character Models for Open Vocabulary Word Representation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1520–1530, Lisbon, Portugal. Association for Computational Linguistics.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W. Black. 2015b. Character-based Neural Machine Translation. *ArXiv e-prints*, November.

Thang Luong, Richard Socher, and Christopher D. Manning. 2013. Better Word Representations with Recursive Neural Networks for Morphology. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning, CoNLL 2013, Sofia, Bulgaria, August 8-9, 2013*, pages 104–113.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015a. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal. Association for Computational Linguistics.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015b. Addressing the Rare Word Problem in Neural Machine Translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 11–19, Beijing, China. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Anoop Deoras, Hai-Son Le, Stefan Kombrink, and Jan Cernocký. 2012. Subword Language Modeling with Neural Networks. Unpublished.

Graham Neubig, Taro Watanabe, Shinsuke Mori, and Tatsuya Kawahara. 2012. Machine Translation without Words through Substring Alignment. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 1: Long Papers*, pages 165–174.

Sonja Nießen and Hermann Ney. 2000. Improving SMT quality with morpho-syntactic analysis. In *18th Int. Conf. on Computational Linguistics*, pages 1081–1085.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pages 1310–1318, Atlanta, USA.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

Rico Sennrich and Barry Haddow. 2015. A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2081–2087, Lisbon, Portugal. Association for Computational Linguistics.

Benjamin Snyder and Regina Barzilay. 2008. Unsupervised Multilingual Learning for Morphological Segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio. Association for Computational Linguistics.

David Stallard, Jacob Devlin, Michael Kayser, Yoong Keok Lee, and Regina Barzilay. 2012. Unsupervised Morphology Rivals Supervised Morphology for Arabic MT. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference, July 8-14, 2012, Jeju Island, Korea - Volume 2: Short Papers*, pages 322–327.

Miloš Stanojević, Amir Kamran, Philipp Koehn, and Ondřej Bojar. 2015. Results of the WMT15 Metrics Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 256–273, Lisbon, Portugal. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014*, pages 3104–3112, Montreal, Quebec, Canada.

Jörg Tiedemann. 2009. Character-based PSMT for Closely Related Languages. In *Proceedings of 13th Annual Conference of the European Association for Machine Translation (EAMT'09)*, pages 12–19.

Jörg Tiedemann. 2012. Character-Based Pivot Translation for Under-Resourced Languages and Domains. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 141–151, Avignon, France. Association for Computational Linguistics.

David Vilar, Jan-Thorsten Peter, and Hermann Ney. 2007. Can We Translate Letters? In *Second Workshop on Statistical Machine Translation*, pages 33–39, Prague, Czech Republic. Association for Computational Linguistics.

Sami Virpioja, Jaakko J. Väyrynen, Mathias Creutz, and Markus Sadeniemi. 2007. Morphology-Aware Statistical Machine Translation Based on Morphs Induced in an Unsupervised Manner. In *Proceedings of the Machine Translation Summit XI*, pages 491–498, Copenhagen, Denmark.

Matthew D. Zeiler. 2012. ADADELTA: An Adaptive Learning Rate Method. *CoRR*, abs/1212.5701.