A Call for Prudent Choice of Subword Merge Operations in Neural Machine Translation

Shuoyang Ding† **Adithya Renduchintala**† **Kevin Duh**†‡
† Center for Language and Speech Processing
‡ Human Language Technology Center of Excellence
Johns Hopkins University
{dings, adi.r}@jhu.edu kevinduh@cs.jhu.edu

Abstract

Most neural machine translation systems are built upon subword units extracted by methods such as Byte-Pair Encoding (BPE) or wordpiece. However, the choice of *number of merge operations* is generally made by following existing recipes. In this paper, we conduct a systematic exploration on different numbers of BPE merge operations to understand how it interacts with the model architecture, the strategy to build vocabularies and the language pair. Our exploration could provide guidance for selecting proper BPE configurations in the future. Most prominently: we show that for LSTM-based architectures, it is necessary to experiment with a wide range of different BPE operations as there is no typical optimal BPE configuration, whereas for Transformer architectures, smaller BPE size tends to be a typically optimal choice. We urge the community to make prudent choices with subword merge operations, as our experiments indicate that a sub-optimal BPE configuration alone could easily reduce the system performance by 3–4 BLEU points.

1 Introduction

While achieving state-of-the-art results, it is a common constraint that Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015; Vaswani et al., 2017) systems are only capable of generating a closed set of symbols. Systems with large vocabulary sizes are too hard to fit onto GPU for training, as the word embedding is generally the most parameter-dense component in the NMT architecture. For that reason, subword methods, such as Byte-Pair Encoding (BPE) (Sennrich et al., 2016), are very widely used for building NMT systems. The general idea of these methods is to exploit the pre-defined vocabulary space optimally by performing a minimum amount of word segmentations in the training set.

However, very few existing literature carefully examines what is the best practice regarding application of subword methods. As hyper-parameter search is expensive, there is a tendency to simply use existing recipes. This is especially true for the *number of merge operations* when people are using BPE, although this configuration is closely correlated with the granularity of the segmentation on the training corpus, thus having direct influence on the final system performance. Prior to this work, Denkowski and Neubig (2017) recommended 32k BPE merge operation in their work on trustable baselines for NMT, while Cherry et al. (2018) contradicted their study by showing that character-based models outperform 32k BPE. Both of these studies are based on the LSTM-based architectures (Sutskever et al., 2014; Bahdanau et al., 2015; Luong et al., 2015). To the best of our knowledge, there is no work that looks into the same problem for the Transformer architecture extensively.[1]

In this paper, we aim to provide guidance for this hyper-parameter choice by examining the interaction between MT system performance with the choice of BPE merge operations under the *low-*

[1] For reference, the original Transformer paper by Vaswani et al. (2017) used BPE merge operations that resulted in 37k joint vocabulary size.

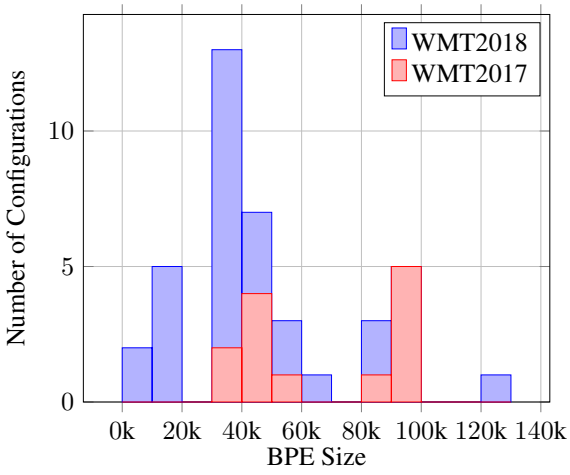


Figure 1: Histogram of BPE merge operations used for in WMT papers from 2017-2018.

resource setting. We conjecture that lower resource systems will be more prone to the performance variance introduced by this choice, and the effect might vary with the choice of model architectures and languages. To verify this, we conduct experiments with 5 different architecture setup on 4 language pairs of IWSLT 2016 dataset. In general, we discover that there is no typical optimal choice of merge operations for LSTM-based architectures, but for Transformer architectures, the optimal choice lays between 0–4k, and systems using the traditional 32k merge operations could lose as much as 4 points in BLEU score compared to the optimal choice.

2 Related Work

Currently, the most common subword methods are BPE (Sennrich et al., 2016), wordpiece (Wu et al., 2016) and subword regularization (Kudo, 2018). Subword regularization introduces Bayesian sampling method to incorporate more segmentation variety into the training corpus, thus improving the systems' ability to handle segmentation ambiguity. Yet, the effect of such method is not very thoroughly tested. In this work we will focus on the BPE/wordpiece method. Because the two methods are very similar, throughout the rest of the paper, we will refer to the BPE/wordpiece method as *BPE method* unless otherwise specified.

To the best of our knowledge, no prior work systematically reports findings for a wide range of systems that cover different architectures and both directions of translation for multiple language pairs. While some work has conducted experiments with different BPE settings, they are generally very limited in the range of configurations explored. For example, Sennrich et al. (2016), the original paper that proposed the BPE method, compared the system performance when using 60k separate BPE and 90k joint BPE. They found 90k to work better and used that for their subsequent winning WMT 2017 new translation shared task submission (Sennrich et al., 2017). Wu et al. (2016), on the other hand, found 8k–32k merge operations achieving optimal BLEU score performance for the wordpiece method. Denkowski and Neubig (2017) explored several hyperparameter settings, including number of BPE merge operations, to establish strong baseline for NMT on LSTM-based architectures. While Denkowski and Neubig (2017) showed that BPE models are clearly better than word-level models, their experiments on 16k and 32k BPE configuration did not show much difference. They therefore recommended "32K as a generally effective vocabulary size and 16K as a contrastive condition when building systems on less than 1 million parallel sentences". However, while studying deep character-based LSTM-based translation models, Cherry et al. (2018) also ran experiments for BPE configurations between 0–32k, and found that the system performance deteriorates with the increasing number of BPE merge operations. Recently, Renduchintala et al. (2018) also showed that it is important to tune the number of BPE merge operations and found no typical optimal BPE configuration for their LSTM-based architecture while sweeping over several language pairs in the low-resource setting. It should be noticed that the results from the above studies actually contradict with each other, and there is still no clear consensus as to what is the best practice for BPE application. Moreover, all the work surveyed above was done with LSTM-based architectures. To this day, we are not aware of any work that explored the interaction of BPE with the Transformer architecture.

To give the readers a better landscape of the current practice, we gather all 44 papers that have been accepted by the research track of Conference of Machine Translation (WMT) through 2017 and 2018. We count different configurations used in a single paper as separate data points. Hence, after removing 8 papers for which BPE is irrelevant, we still manage to obtain 42 data points, shown in Figure 1. It first comes to our attention that 30k–40k is the most popular range for the number of BPE merge operations. This is mostly driven

by the popularity of two configurations: 30k and 32k. 80k–100k is also pretty popular, which is largely due to configurations 89.5k and 90k. Upon closer examination, we realized that most papers that used 90k were following the configuration in Sennrich et al. (2017), the winning NMT system in the WMT 2017 news translation shared task, but this setup somehow became less popular in 2018. On the other hand, although we are unable to confirm a clear trend-setter, 30k–50k always seems to be a common choice. Moreover, although smaller BPE size got more popular among configurations in 2018, none of the work published in WMT has ever explored BPE size lower than 6k. All of the above observations support our initial claim that we as a community have not yet systematically investigated the entire range of BPE merge operations used in our experiments.

3 Analysis Setup

Our goal is to compare the impact of different numbers of BPE merge operations on multiple language pairs and multiple NMT architectures. We experiment with the following BPE merge operation setup: 0 (character-level), 0.5k, 1k, 2k, 4k, 8k, 16k, and 32k, on both translation directions of 4 language pairs and 5 architectures. Additionally, we include 6 more language pairs (with 2 architectures) to study the interaction between linguistic attributes and BPE merge operations.

3.1 Dataset

Our experiments are conducted with the all the data from IWSLT 2016 shared task, covering translation of English (en) from and into Arabic (ar), Czech (cs), French (fr) and German (de). As this dataset contains multiple dev and test sets, we concatenate all the dev sets into a single dev set and do the same for the test set as well. To increase language coverage, we also conduct extra experiments with 6 more language pairs from the TED corpus (Qi et al., 2018). We use Brazilian Portuguese (pt), Hebrew (he), Russian (ru), Turkish (tr), Polish (pl) and Hungarian (hu) as our extra languages, paired with English. All the data are tokenized and truecased using the accompanying script from Moses decoder (Koehn et al., 2007) before training and applying BPE models.[2]

We use subword-nmt[3] to train and apply BPE to our data. Unless otherwise specified, all of our BPE models are trained on the concatenation of the source and target training corpus, i.e. the *joint BPE* scheme in Sennrich et al. (2016). We use SacreBLEU (Post, 2018) to compute BLEU score.[4]

3.2 Architecture

We build our NMT system with fairseq (Ott et al., 2019). We use two pre-configured architectures in fairseq for our study, namely `lstm-wiseman-iwslt-de-en` (referred to as `tiny-lstm`) and `transformer-iwslt-de-en` (referred to as `deep-transformer`), which are the model architecture tuned for their benchmark system trained on IWSLT 2014 German-English data. However, we find (as can be seen from Table 1) that the number of parameters in `lstm-tiny` is a magnitude lower than `deep-transformer` mainly due to the fact that the former has a single-layer uni-directional encoder and a single-layer decoder, while the later has 6 encoder and decoder layers. For a fairer comparison we include a `deep-lstm` architecture with 6 encoder and decoder layers which roughly matches the number of parameters in `deep-transformer`. To study the effect of BPE on relatively smaller architectures, we also include `shallow-transformer` and `shallow-lstm` architectures, both with 2 encoder and decoder layers. The `shallow-lstm` also use bidirectional LSTM layers in the encoder. These two architectures also roughly match each other in terms of number of parameters. With these 5 architectures, we believe we have covered a wide range of common choices in NMT architectures, especially in low-resource settings. We use Adam optimizer (Kingma and Ba, 2014) for all the experiments we run. For Transformer experiments, we use the learning rate scheduling settings in Vaswani et al. (2017), including the inverse square root learning rate scheduler, 4000 warmup updates and initial warmup learning rate of 1×10^{-7}. For most LSTM experiments, we just use learning rate 0.001 from the start and reduce the learning rate by half every time the loss function fails to improve on the development set. However, we find that for `deep-lstm` architecture, such learning rate schedule tends to be unstable, which is very similar to training Transformer without the warmup

[2] Data processing scripts available at `https://github.com/shuoyangd/prudent-bpe`.

[3] `https://pypi.org/project/subword-nmt/0.3.5/`

[4] SacreBLEU signature:`BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.2.12`.

	bi-dir	d_{enc}	d_{dec}	d_{emb}	l	N_h	N_p
shallow-transformer	N/A	512	512	512	2	4	18.8M
deep-transformer	N/A	512	512	512	6	4	39.8M
tiny-lstm	no	256	256	256	1	1	5.6M
shallow-lstm	yes	384	384	384	2	1	16.4M
deep-lstm	yes	384	384	384	6	1	35.3M

Table 1: Information of the 5 architectures used for analysis. **bi-dir** is a boolean representing whether the encoder is bi-directional. d_{enc}, d_{dec} and d_{emb} are dimension of encoder, decoder and source/target word embedding, respectively. l is the number of encoder/decoder layers. N_h is the number of attention heads, while N_p is the number of parameters of the model at 8k BPE merge operations.

learning rate schedule. Applying the same warmup schedule as Transformer experiments works for most `deep-lstm` architecture except for de-en experiments as BPE size 16k and 32k, for which we have to apply 8000 warmup updates. Per the experiment setting in Vaswani et al. (2017), we also apply label smoothing with $\varepsilon_{ls} = 0.1$ for all of our Transformer experiments.

4 Analysis

4.1 Analysis 1: Architectures

Table 2 shows the BLEU score for Transformer systems with BPE merge operations ranging from 0 to 32k. The Transformer experiments show a clear trend; large BPE settings of 16k-32k are *not* optimal for low-resource settings. We see that regardless of the direction of translation, the best BLEU score for Transformer-based architectures are somewhere in the 0-1k range. Although there is not much drop for 2k-4k, there is generally a drastic performance drop as the number of BPE merge operation is increased beyond 8k. It should also be noted that the difference between the best and the worst performance is around 3 BLEU points (refer to the δ column in Table 2), larger than the improvements claimed in many machine translation papers.

Table 3 shows the BLEU score for LSTM-based architectures trained with BPE merge operations ranging from 0 to 32k. Among the three tables, the `shallow-lstm` architecture has the minimal variation with regard to different merge operation choices. For `tiny-lstm`, we observe a drastic performance drop between BPE merge operations 0/500 or 500/1k. But aside from these two settings, the variation is of similar scale to `shallow-lstm`. For `deep-lstm`, the variation is even larger than the Transformer architectures, and compared to `tiny-lstm` and `shallow-lstm`, the optimal BPE configuration shifts to BPE sizes on the smaller end. However, we have also noticed that the overall absolute BLEU score of `deep-lstm` is lower than `shallow-lstm` despite more parameter is being used. We conjecture that the larger variation and lower BLEU score from the `deep-lstm` experiments is largely due to the overfitting effect on the small training data. Despite this effect, moving from tiny to deep model, we observe a trend that deeper models tends to make use of smaller BPE size better. In general, we conclude that unlike Transformer architecture, there is no typical optimal BPE configuration setting for the LSTM architecture. Because of this noisiness, we urge that future work using LSTM-based baselines tune their BPE configuration in a wider range on a development set to the extent possible, in order to ensure reasonable comparison.

4.2 Analysis 2: Joint vs Separate BPE

Another question that is not extensively explored in the existing literature is whether *joint BPE* is the definitive better approach to apply BPE. The alternative way, referred to here as separate BPE, is to build separate models for source and target side of the parallel corpus. Sennrich et al. (2016) conducted experiments with both joint and separate BPE, but these experiments were conducted with different BPE size, and not much analysis was conducted on the separate BPE model. Huck et al. (2017) is the only other work we are aware of that used with separate BPE models for their study. It was mentioned that their joint BPE vocabulary of 59500 yielded a German vocabulary twice as large as English, which is an undesirable characteristic for their study.

Before comparing the system performance, we would like to systematically understand how the resulting vocabulary is different when jointly and separately applying BPE. Table 4 shows the two

		0	0.5k	1k	2k	4k	8k	16k	32k	δ
deep-transformer	**ar-en**	30.3	30.8	30.6	30.5	30.4	29.8	28	27.5	3.3
	cs-en	24.6	23.3	23.0	22.7	21.2	22.6	20.6	21.0	4.0
	de-en	28.1	28.6	28.0	28.4	27.7	27.5	26.7	25.2	3.4
	fr-en	28.8	29.8	29.6	29.3	28.7	28.5	27.5	26.6	3.2
	en-ar	12.6	13.0	12.1	12.3	11.8	11.3	10.7	10.6	2.4
	en-cs	17.3	17.1	16.7	16.4	16.1	15.6	14.7	13.8	3.5
	en-de	26.1	27.4	27.4	26.1	26.3	26.1	25.8	23.9	3.5
	en-fr	25.2	25.6	25.3	25.5	25.3	24.7	24.1	22.8	2.8
shallow-transformer	**ar-en**	26.4	27.9	28.7	28.5	28.6	27.7	26.2	25.5	3.2
	cs-en	22.4	22.6	22.3	21.8	21.7	21.1	21.1	20.1	2.5
	de-en	25.5	27.4	27.1	27.3	27.1	25.9	24.6	23.7	3.7
	fr-en	26.3	28.0	28.9	28.0	28.0	27.4	26.1	26.1	2.7
	en-ar	11.7	11.2	11.5	11.0	11.3	10.5	9.5	9.0	2.7
	en-cs	16.4	16.7	16.0	16.2	14.4	14.2	13.9	13.9	2.8
	en-de	23.8	25.7	25.4	25.3	25.2	24.3	24.1	22.1	3.6
	en-fr	23.5	24.7	25.1	24.6	24.5	23.8	22.7	22.1	3.0

Table 2: BLEU score for Transformer architectures with multiple BPE configurations. Each score is color-coded by its rank among scores from different BPE configurations in the same row. δ is the difference between the best and worst BLEU score of each row.

		0	0.5k	1k	2k	4k	8k	16k	32k	δ
tiny-lstm	**ar-en**	20.6	22.1	22.4	23.0	24.1	24.2	24.2	24.0	3.6
	cs-en	17.8	19.1	18.8	19.0	19.2	19.5	20.7	19.1	2.9
	de-en	21.1	22.5	23.2	23.1	23.1	23.1	23.6	23.0	2.5
	fr-en	21.8	25.3	25.3	25.4	25.1	25.3	25.1	24.7	3.6
	en-ar	8.5	8.7	9.3	8.8	8.8	8.6	8.8	8.8	0.8
	en-cs	11.5	12.3	13.7	13.2	13.0	14.1	14.4	13.2	2.9
	en-de	18.2	20.8	21.4	21.1	21.9	21.6	21.0	21.6	3.7
	en-fr	19.9	20.4	20.7	21.8	21.3	21.0	21.3	21.3	1.7
shallow-lstm	**ar-en**	27.5	27.2	27.1	27.6	27.4	26.7	27.5	26.3	1.3
	cs-en	22.2	22.2	22.2	22.9	22.7	23.0	22.8	21.6	1.4
	de-en	25.7	25.9	26.0	25.9	26.4	26.3	26.1	26.5	0.8
	fr-en	27.6	26.7	27.7	28.4	27.9	27.7	28.5	27.5	1.8
	en-ar	11.0	11.0	10.7	10.4	10.6	10.6	10.4	10.1	0.9
	en-cs	16.1	15.7	15.8	15.3	15.8	15.5	15.8	15.6	0.8
	en-de	24.9	25.1	23.9	24.2	25.4	25.2	25.5	25.0	1.6
	en-fr	24.3	23.8	23.7	24.2	23.5	24.1	23.9	23.0	1.3
deep-lstm	**ar-en**	21.2	25.7	27.2	27.1	25.6	24.8	25.1	22.9	4.3
	cs-en	19.8	22.0	18.5	21.1	20.9	21.2	20.3	15.8	6.2
	de-en	25.7	25.2	24.9	24.1	24.5	23.5	23.5	23.1	2.6
	fr-en	25.6	26.8	27.1	26.0	26.9	25.6	17.9	22.8	9.2
	en-ar	10.9	10.2	10.3	7.5	9.5	9.4	7.2	8.0	3.7
	en-cs	13.7	14.6	15.3	14.6	12.2	12.6	11.9	12.6	3.4
	en-de	22.4	24.9	23.6	23.9	22.4	24.0	24.3	23.4	2.5
	en-fr	23.1	22.9	23.5	23.1	22.2	22.0	18.0	20.0	5.5

Table 3: BLEU score for LSTM architectures with multiple BPE configurations. Each score is color-coded by its rank among scores from different BPE configurations in the same row. δ is the difference between the best and worst BLEU score of each row.

		Char	Separate BPE			Joint BPE		
			2k	8k	32k	2k	8k	32k
ar-en	**src**	0.49k	2.48k	8.47k	32.36k	2.46k	7.98k	26.11k
	tgt	0.24k	2.23k	8.17k	30.45k	1.27k	4.06k	13.45k
fr-en	**src**	0.30k	2.30k	8.26k	31.23k	2.18k	7.14k	24.48k
	tgt	0.23k	2.22k	8.16k	30.40k	1.94k	6.10k	20.45k

Table 4: Vocabulary size after applying separate and joint BPE for ar-en and fr-en language pair.

		Best Sep.	Best Joint	Worst Sep.	Worst Joint
tiny-lstm	**ar-en**	24.3	24.2	20.6	20.6
	cs-en	20.2	20.7	17.8	17.8
	de-en	23.3	23.6	21.1	21.1
	fr-en	25.0	25.4	21.8	21.8
	en-ar	9.1	9.3	8.3	8.5
	en-cs	15.2	14.4	11.5	11.5
	en-de	21.8	21.9	18.2	18.2
	en-fr	21.1	21.8	19.9	19.9
deep-transformer	**ar-en**	31.0	30.8	26.8	27.5
	cs-en	24.6	24.6	19.0	20.6
	de-en	28.1	28.6	24.8	25.2
	fr-en	28.8	29.8	27.3	26.6
	en-ar	12.0	13.0	9.6	10.6
	en-cs	17.3	17.3	13.0	13.8
	en-de	27.3	27.4	23.8	23.9
	en-fr	24.0	25.6	22.5	22.8

Table 5: Best and worst BLEU score with `tiny-lstm` and `deep-transformer` for joint and separate BPE models.

most typical cases for this comparison, namely the Arabic-English language pair and the French-English language pair. The reason these two language pairs are typical is that for Arabic-English, the scripts of the two languages are completely different, while the French and English scripts only have minor difference. It could be seen that for Arabic-English language pair, the Arabic vocabulary size is always roughly twice the size of the English vocabulary. Upon closer examination, we see that roughly half of the Arabic vocabulary is consisted of English words and subwords, scattering over around 2% of the lines in the Arabic side of the training corpus.[5] Hence, for most sentence pairs in the training data, the *effective* Arabic and English vocabulary under joint BPE model is still roughly the same size. On the other hand, because of extensive subword vocabulary sharing, at lower BPE size, the vocabulary size for French and English is always roughly the same as the number of BPE merge operations regardless of separate or joint BPE. However, this equality starts to diverge as more BPE merge operations are conducted, because the vocabulary difference between French and English starts to play out in this scenario. Unlike Arabic-English, it is hard to predict what is the resulting BPE size from the number of merge operations used, because it is hard to know how many resulting subwords will be shared between the two languages.

Table 5 shows our experimental results with separate/joint BPE and our base architectures.[6] With the configurations we explore, the difference between the best separate/joint BPE performance seems minimal. On the other hand, while the worst BPE configuration remains the same for separate BPE models, we see even worse performance for Transformer at 32k separate BPE most of the time. We think this is a continuation of the trend observed in our main results, as the vocabulary size tends to be even larger than joint BPE when applying separate BPE models.

Given the negligible difference in model performance, we think it is not necessary to sweep BPE merge operations for both joint and separate settings. It is sufficient to focus on the setting that makes the most sense for the task at hand, and focus on hyper parameter search within that setting.

4.3 Analysis 3: Languages

We are interested in what properties of the language have the most impact on the variance of BLEU score with regard to different BPE configurations. For our main experiments, we can already see a pretty consistent trend that for `deep-transformer` architecture, 0.5k and 32k merge operations always roughly correspond to the best and worst BPE configurations, respectively.

[5]These English tokens are generally English names, URLs or other untranslated concepts or acronyms.

[6]We only run experiments on 2k, 8k and 32k to save computation time.

	0.5k	32k	δ		0.5k	32k	δ
pt-en	36.3	34.7	1.6	**en-pt**	38.5	35.6	2.9
he-en	31.1	28.6	2.5	**en-he**	26.2	22.9	3.3
tr-en	20.9	17.8	3.1	**en-tr**	13.0	9.8	3.2
ru-en	19.9	18.0	1.9	**en-ru**	19.1	16.6	2.5
pl-en	19.3	16.7	2.6	**en-pl**	16.7	13.4	3.3
hu-en	20.8	16.8	4.0	**en-hu**	16.0	12.6	3.4

Table 6: BLEU score for the 6 extra language pairs in multilingual-TED dataset with `deep-transformer` architecture.

	coef.	std. error	p-value
f_1	0.575	1.345	0.677
f_2	-0.460	1.345	0.738
f_3	-1.998	1.983	0.333
f_4	0.304	0.360	0.415
f_5	1.060	0.639	0.123
f_6	1.169	0.516	0.043
f_7	0.913	0.314	0.013
f_8	0.340	0.367	0.373
f_9	1.280	0.755	0.116

Table 7: Coefficient from regression analysis and their corresponding standard error and p-values. f_1 and f_2 are source and target type/token ratio, respectively. f_3 is alignment ratio. f_4–f_6 are binary features for source-side morphological type (fusional, introflexive and agglutinative) and f_7–f_9 are the same for target.

To add more data points, we assume 0.5k and 32k are always the best and the worst configurations and build systems with these two configurations with both translation directions of 6 more languages pairs, namely, translating of English into and out of Brazilian Portuguese (pt), Hebrew (he), Russian (ru), Turkish (tr), Polish (pl) and Hungarian (hu). Table 6 shows the result with these 6 language pairs. We note that our observation for the 4 language pairs generalize well for the extra 6 language pairs, and we observe a similar magnitude of performance drop as the other language pairs moving from 0.5k to 32k.

To acquire insights for the aforementioned problem, we conduct a linear regression analysis using the linguistic features of the the 10 language pairs as independent variables and BLEU score difference between 0.5k and 32k merge operation settings as the dependent variable.[7] The linguistic features of our interest are described as follows:

- **Type/Token Ratio**: Taken from Bentz et al. (2016) this is the ratio between number of token types and the number of tokens in the training corpus, ranging $[0, 1]$. These are computed separately for source and target language and denoted as f_1 and f_2 respectively.
- **Alignment Ratio**: Also taken from Bentz et al. (2016), this is the relative difference between the number of many-to-one alignments and one-to-many alignments in the training corpus, ranging $[-1, 1]$. We follow the same alignment setting as in Renduchintala et al. (2018). This is computed together for each parallel training corpus and denoted as f_3.
- **Morphological Type**: We then use a set of binary features to indicate if a language exhibits a certain morphological patterns. We take morphological features from Gerz et al. (2018), where for each language a morphological type from the following categories was assigned: *Isolating*, *Fusional*, *Introflexive* and *Agglutinative*. None of the languages we use exhibit *Isolating* morphology which leaves us with 6 binary features. The features f_4, f_5 and f_6 indicates the presence (or absence) of *fusional*, *introflexive* and *agglutinative* morphological patterns respectively for the source language and f_7, f_8, f_9 indicate the same for the target side.

The 9 features are re-normalized to the $[0, 1]$ region with the min-max normalization. Our linear regression analysis is conducted with Ordinary Least Squares (OLS) model in the Python statsmodels[8] package.

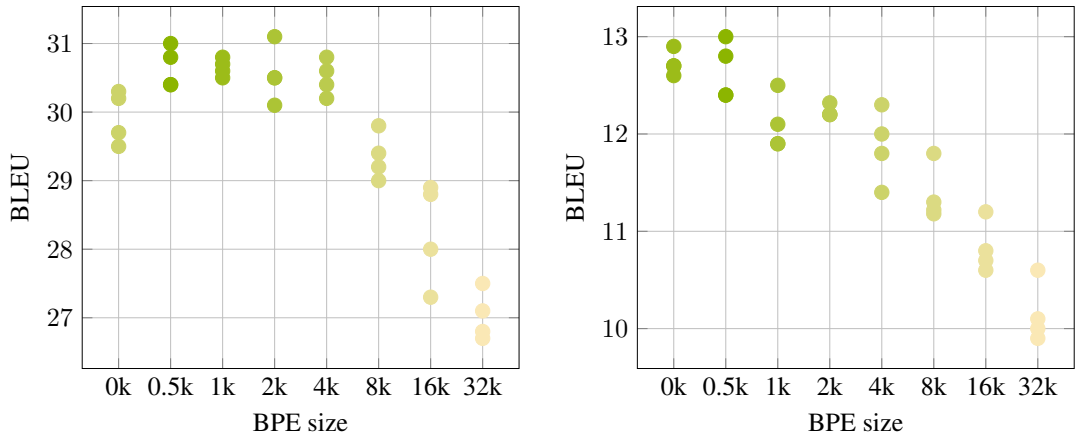
Table 7 shows the regression result. Surprisingly, we don't see any strong correlation between the type/token ratio, alignment ratio and the variance in BPE. On the other hand, the regression points out that having agglutinative language on the source side and fusional language on the target side increases such variance. While we have seen significant BPE variances for all the experiments with Transformer, we think future work should be especially cautious with systems that translate out of agglutinative language and into fusional language (note that English is classified as fusional language in this regime).

4.4 Analysis 4: Variance with Random Seeds

Since our experiments are under low-resource settings, it is important to examine whether the trends

[7] Note that for language pairs in our main results, these may not necessarily the best or the worst system. But the readers shall see that the difference is pretty minimal.

[8] `https://pypi.org/project/statsmodels/0.9.0/`



(a) ar-en (b) en-ar

Figure 2: Scatter plots for the variance analysis of `deep-transformer` system. Each dot in the plot represents the BLEU score for one random restart, while the color code follows the result ranking of its corresponding system configuration in Table 2.

we observe above are due to different system configurations or mostly variance of random seeds. As it is expensive to re-run all the systems multiple times, we only conduct such analysis on the `deep-transformer` architecture and ar-en and en-ar language pairs. We choose to focus on Transformer architecture because we observe more consistent trend for Transformer than LSTM. Hence, it is more interesting to see how well it holds against the randomness in training. To conduct such analysis, we run each system configuration for three more times with different random seeds resulting in four points for each system configuration.

Figure 2 shows the scatter plots of BLEU scores for each random restart under each system configuration. Ideally, the BLEU scores from multiple random restarts of the system configurations should preserve the same ranking as the results in Table 2. It can be seen that, the results from the top-3 BPE configurations are often clustered together (indicating low variance) and the rankings of the other configurations are preserved pretty well. Specifically, even best instances among multiple random restarts with 16k and 32k BPE merge operations fall pretty far from those with top configurations, further verifying our previous observations on the Transformer architecture.

4.5 Analysis 5: High-Resource Setting

While this paper focuses on low-resource settings, we conduct one set of experiments with a high-resource language pair to see if our results generalize to high-resource settings. This experiment is conducted with all WMT 2017 Russian-English (ru-en) data except the UN dataset, which includes 2.61M sentence pairs in total. We use the test sets from news translation shared task of WMT 2012-2016 as the development data and test on WMT 2017 test set. Due to computation constraints, we only experiment with `deep-transformer` architecture. All the other configurations are exactly the same as the low-resource experiments.

Table 8 summarizes the results. First, notice that the overall variance of results under different BPE configurations is relatively smaller than the low-resource experiments, verifying our intuition that it is especially important to tune BPE size under low-resource settings. Besides, the trend in this setting is also very different from what is shown in Table 2. Specifically, the best results are often obtained with larger BPE sizes, which explains why these configurations were preferred by previous analysis. It could hence be concluded that the analysis results in this paper should *not* be generalized to high-source settings. We leave comprehensive analysis with high-resource language pairs for future work.

5 Conclusion

We conduct a systematic exploration over various numbers of BPE merge operations to understand its interaction with system performance. We conduct this investigation over 5 different NMT architectures including encoder-decoder and Transformer, and 4 language pairs in both translation directions. We leave systematic study on the effect of BPE on high-resource settings and more language pairs, especially morphologically isolating languages, for

	0	0.5k	1k	2k	4k	8k	16k	32k	δ
ru-en	29.3	30.4	30.0	30.3	30.6	30.9	31.0	30.9	1.7
en-ru	28.0	29.1	29.1	29.5	29.5	29.8	30.0	30.0	2.0

Table 8: BLEU score for `deep-transformer` architecture under high-resource setting, with multiple BPE configurations. Each score is color-coded by its rank among scores from different BPE configurations in the same row. δ is the difference between the best and worst BLEU score of each row.

future work. Subword regularization could also be studied in this manner.

Based on the findings, we make the following recommendations for selecting BPE merge operations in the future:

- For Transformer-based architectures, we recommend the sweep be concentrated in the $0 - 4$k range.
- For Shallow LSTM architectures, we find no typically optimal BPE merge operation and therefore urge future work to sweep over $0 - 32$k to the extent possible.
- We find no significant performance differences between joint BPE and seperate BPE and therefore recommend BPE sweep be conducted with either of these settings.

Furthermore, we strongly urge that the aforementioned checks be conducted when translating into fusional languages (such as English or French) or when translating from agglutinative languages (such as Turkish).

Our hope is that future work could take the experiments presented here to guide their choices regarding BPE and wordpiece configurations, and that readers of low-resource NMT papers call for appropriate skepticism when the BPE configuration for the experiments appears to be sub-optimal.

Acknowledgments

This work is supported in part by the Office of the Director of National Intelligence, Intelligence Advanced Research Projects Activity (IARPA), via contract #FA8650-17-C-9115. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors.

References

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Bentz, Christian, Tatyana Ruzsics, Alexander Koplenig, and Tanja Samardzic. 2016. A comparison between morphological complexity measures: typological data vs. language corpora. In *Proceedings of the Workshop on Computational Linguistics for Linguistic Complexity (CL4LC)*, pages 142–153.

Cherry, Colin, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting character-based neural machine translation with capacity and compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 4295–4305.

Denkowski, Michael J. and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 18–27.

Gerz, Daniela, Ivan Vulić, Edoardo Maria Ponti, Roi Reichart, and Anna Korhonen. 2018. On the relation between linguistic typology and (limitations of) multilingual language modeling. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 316–327, Brussels, Belgium, October-November. Association for Computational Linguistics.

Huck, Matthias, Simon Riess, and Alexander M. Fraser. 2017. Target-side word segmentation strategies for neural machine translation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 56–67.

Kingma, Diederik P and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koehn, Philipp, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*.

Kudo, Taku. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, pages 66–75.

Luong, Thang, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 1412–1421.

Ott, Myle, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. *arXiv preprint arXiv:1904.01038*.

Post, Matt. 2018. A call for clarity in reporting bleu scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels, October. Association for Computational Linguistics.

Qi, Ye, Devendra Singh Sachan, Matthieu Felix, Sarguna Padmanabhan, and Graham Neubig. 2018. When and why are pre-trained word embeddings useful for neural machine translation? In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 2 (Short Papers)*, pages 529–535.

Renduchintala, Adithya, Pamela Shapiro, Kevin Duh, and Philipp Koehn. 2018. Character-aware decoder for neural machine translation. *CoRR*, abs/1809.02223.

Sennrich, Rico, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Sennrich, Rico, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural MT systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, pages 389–399.

Sutskever, Ilya, 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, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 6000–6010.

Wu, Yonghui, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.