The University of Edinburgh's Submissions to the WMT19 News Translation Task

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, Alexandra Birch

The University of Edinburgh's Submissions to the WMT19 News Translation Task

Rachel Bawden **Nikolay Bogoychev** **Ulrich Germann** **Roman Grundkiewicz**
Faheem Kirefu **Antonio Valerio Miceli Barone** **Alexandra Birch**

School of Informatics, University of Edinburgh, Scotland
rachel.bawden@ed.ac.uk

Abstract

The University of Edinburgh participated in the WMT19 Shared Task on News Translation in six language directions: English↔Gujarati, English↔Chinese, German→English, and English→Czech. For all translation directions, we created or used back-translations of monolingual data in the target language as additional synthetic training data. For English↔Gujarati, we also explored semi-supervised MT with cross-lingual language model pre-training, and translation pivoting through Hindi. For translation to and from Chinese, we investigated character-based tokenisation vs. sub-word segmentation of Chinese text. For German→English, we studied the impact of vast amounts of back-translated training data on translation quality, gaining a few additional insights over Edunov et al. (2018). For English→Czech, we compared different pre-processing and tokenisation regimes.

1 Introduction

The University of Edinburgh participated in the WMT19 Shared Task on News Translation in six language directions: English-Gujarati (EN↔GU), English-Chinese (EN↔ZH), German-English (DE→EN) and English-Czech (EN→CS). All our systems are neural machine translation (NMT) systems trained in constrained data conditions with the Marian[1] toolkit (Junczys-Dowmunt et al., 2018). The different language pairs pose very different challenges, due to the characteristics of the languages involved and arguably more importantly, due to the amount of training data available.

Pre-processing For EN↔ZH, we investigate character-level pre-processing for Chinese compared with subword segmentation. For EN→CS, we show that it is possible in high resource settings to simplify pre-processing by removing steps.

Exploiting non-parallel resources For all language directions, we create additional, synthetic parallel training data. For the high resource language pairs, we look at ways of effectively using large quantities of backtranslated data. For example, for DE→EN, we investigated the most effective way of combining genuine parallel data with larger quantities of synthetic parallel data and for CS→EN, we filter backtranslated data by re-scoring translations using the MT model for the opposite direction. The challenge for our low resource pair, EN↔GU, is producing sufficiently good models for back-translation, which we achieve by training semi-supervised MT models with cross-lingual language model pre-training (Lample and Conneau, 2019). We use the same technique to translate additional data from a related language, Hindi.

NMT Training settings In all experiments, we test state-of-the-art training techniques, including using ultra-large mini-batches for DE→EN and EN↔ZH, implemented as optimiser delay.

Results summary Official automatic evaluation results for all final systems on the WMT19 test set are summarised in Table 1. Throughout the paper, BLEU is calculated using SACREBLEU[2] (Post, 2018) unless otherwise indicated. Our final EN-GU models are available for download.[3,4]

2 Gujarati ↔ English

One of the main challenges for translation between English↔Gujarati is that it is a low-resource language pair; there is little openly available parallel data and much of this data is domain-specific

[1] https://marian-nmt.github.io

[2] https://github.com/mjpost/sacreBLEU

[3] See data.statmt.org/wmt19_systems/ for our released EN-GU models and running scripts.

[4] Note that following the discovery of a pre-processing error, the EN→GU and GU→EN models have been retrained and achieve BLEU scores of 16.3 and 22.3 respectively.

Lang. direction	BLEU	Ranking
EN→GU	16.4	1
GU→EN	21.4	2
EN→ZH	34.4	7
ZH→EN	27.7	6
DE→EN	35.0	9
EN→CS	27.9	3

Table 1: Final BLEU score results and system rankings amongst constrained systems according to automatic evaluation metrics.

and/or noisy (cf. Section 2.1). Our aim was therefore to experiment how additional available data can help us to improve translation quality: large quantities of monolingual text for both English and Gujarati, and resources from Hindi (a language related to Gujarati) in the form of monolingual Hindi data and a parallel Hindi-English corpus. We applied semi-supervised translation, backtranslation and pivoting techniques to create a large synthetic parallel corpus from these resources (Section 2.2), which we used to augment the small available parallel training corpus, enabling us to train our final supervised MT models (Section 2.3).

2.1 Data and pre-processing

We trained our models using only data listed for the task (cf. Table 2). Note that we did not have access to the corpora provided by the Technology Development for Indian Languages Programme, as they were only available to Indian citizens.

We pre-processed all data using standard scripts from the Moses toolkit (Koehn et al., 2007): normalisation, tokenisation, cleaning (of training data only, with a maximum sentence length of 80 tokens) and true-casing for English data, using a model trained on all available news data. The Gujarati data was additionally pre-tokenised using the IndicNLP tokeniser[5] before Moses tokenisation was applied. We also applied subword segmentation using BPE (Sennrich et al., 2016b), with joint subword vocabularies. We experimented with different numbers of BPE operations during training.

2.2 Creation of synthetic parallel data

Data augmentation techniques such as backtranslation (Sennrich et al., 2016a; Edunov et al., 2018), which can be used to produce additional synthetic parallel data from monolingual data, are standard in MT. However they require a sufficiently good intermediate MT model to produce translations that are of reasonable quality to be useful for training (Hoang et al., 2018). This is extremely hard to achieve for this language pair. Our preliminary attempt at parallel-only training yielded a very low BLEU score of 7.8 on the GU→EN development set using a Nematus-trained shallow RNN with heavy regularisation,[6] and similar scores were found for a Moses phrase-based translation system.

Our solution was to train models for the creation of synthetic data that exploit both monolingual and parallel data during training.

Lang(s)	Corpus	#sents	Ave. len.
	Parallel data		
EN-GU	Software data	107,637	7.0
	Wikipedia	18,033	21.1
	Wiki titles v1	11,671	2.1
	Govin	10,650	17.0
	Bilingual dictionary	9,979	1.5
	Bible	7,807	26.4
	Emille	5,083	19.1
GU-HI	Emille	7,993	19.1
EN-HI	Bombay IIT	1.4M	13.4
	Monolingual data		
EN	News	200M	23.6
GU	Common crawl	3.7M	21.9
	Emille	0.9M	16.6
	Wiki-dump	0.4M	17.7
	News	0.2M	15.4
HI	Bombay IIT	45.1M	18.7
	News	23.6M	17.0

Table 2: EN-GU Parallel training data used. Average length is calculated in number of tokens per sentence. For the parallel corpora, this is calculated for the first language indicated (i.e. EN, GU, then EN)

2.2.1 Semi-supervised MT with cross-lingual language model pre-training

We followed the unsupervised training approach in (Lample and Conneau, 2019) to train two MT systems, one for EN↔GU and a second for HI→GU.[7] This involves training unsupervised NMT models with an additional supervised MT training step. Initialisation of the models is done by pre-training parameters using a masked language modelling objective as in Bert (Devlin et al., 2019), individually for each language (MLM, which stands for *masked language modelling*) and/or cross-lingually

[5] anoopkunchukuttan.github.io/indic_nlp_library/

[6] Learning rate: 5×10^{-4}, word dropout (Gal and Ghahramani, 2016): 0.3, hidden state and embedding dropout: 0.5, batch tokens: 1000, BPE vocabulary threshold 50, label smoothing: 0.2.

[7] We used the code available at https://github.com/facebookresearch/XLM

(TLM, which stands for *translation language modelling*). The TLM objective is the MLM objective applied to the concatenation of parallel sentences. See (Lample and Conneau, 2019) for more details.

2.2.2 EN and GU backtranslation

We trained a single MT model for both language directions EN→GU and GU→EN using this approach. For pre-training we used all available data in Table 2 (both the parallel and monolingual datasets) with MLM and TLM objectives. The same data was then used to train the semi-supervised MT model, which achieved a BLEU score of 22.1 for GU→EN and 12.6 for EN→GU on the dev set (See the first row in Table 5). This model was used to backtranslate 7.3M of monolingual English news data into Gujarati and 5.1M monolingual Gujarati sentences into English.[8]

System and training details We use default architectures for both pre-training and translation: 6 layers with 8 transformer heads, embedding dimensions of 1024. Training parameters are also as per the default: batch size of 32, dropout and attention dropout of 0.1, Adam optimisation (Kingma and Ba, 2015) with a learning rate of 0.0001.

Degree of subword segmentation We tested the impact of varying degrees of subword segmentation on translation quality (See Figure 1). Contrary to our expectation that a higher degree of segmentation (i.e. with a very small number of merge operations) would produce better results, as is often the case with very low resource pairs, the best tested value was 20k joint BPE operations. The reason for this could be the extremely limited shared vocabulary between the two languages[9] or that training on large quantities of monolingual data turns the low resource task into a higher one.

2.2.3 HI→GU translation

Transliteration of Hindi to Gujarati script We first transliterated all of the Hindi characters into Gujarati characters to encourage vocabulary sharing. As there are slightly more Hindi unicode characters than Gujarati, Hindi characters with no corresponding Gujarati characters and all non-Hindi characters were simply copied across.

Once transliterated, there is a high degree of overlap between the transliterated Hindi (HG) and the corresponding Gujarati sentence, which is demonstrated by the example in Figure 2.

[8]We were unable to translate all available monolingual data due to time constraints and limits to GPU resources.

[9]Except for occasional Arabic numbers and romanised proper names in Gujarati texts.

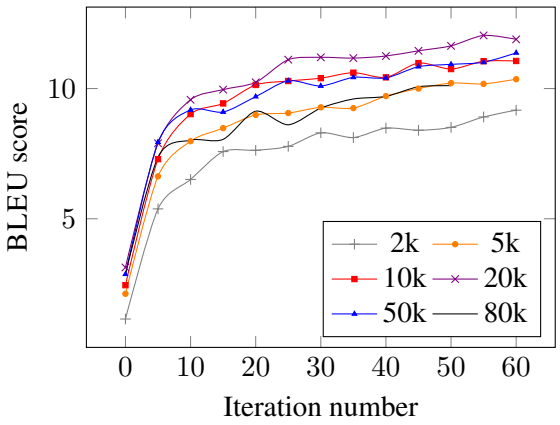

Figure 1: The effect of the number of subword operations on BLEU score during training for EN→GU (calculated on the *newsdev2019* dataset).

Our parallel Gujarati-Hindi data consisted of approximately 8,000 sentences from the Emille corpus. After transliterating the Hindi, we found that 9% of Hindi tokens (excluding punctuation and English words) were an exact match to the corresponding Gujarati tokens. However, we did have access to large quantities of monolingual data in both Gujarati and Hindi (see Table 2), which we pre-processed in the same way.

The semi-supervised HI↔GU system was trained using the MLM pre-training objective described in Section 2.1 and the same model architecture as the EN↔GU model in Section 2.2.2. For the MT step, we trained on 6.5k parallel sentences, reserving the remaining 1.5k as a development set. As with the EN↔GU model, we investigated the effect of different BPE settings (5k, 10k, 20k and 40k merge operations) on the translation quality. Surprisingly, just as with EN↔GU, 20k BPE operations performed best (cf. Table 3), and so we used the model trained in this setting to translate the Hindi side of the IIT Bombay English-Hindi Corpus, which we refer to as HI2GU-EN.

BPE	5k	10k	20k	40k
BLEU	15.4	16.0	16.3	14.6

Table 3: The influence of number of BPE merge operations on HI→GU BLEU score measured using BLEU scores on the development set

GU: એમને સાવધાનીપૂર્વક સાફ કરો અને દન્ત ચિકિત્સક ની જોડે નિયમિત જાવો .
HI: उनको सावधानीपूर्वक साफ करें और दन्त चिकित्सक के पास नियमित जायें ।
HG: ઉનકો સાવધાનીપૂર્વક સાફ કરેં ઔર દન્ત ચિકિત્સક કે પાસ નિયમિત જાયેં .
Gloss: THEM CAREFULLY CLEAN DO AND TEETH DOCTOR POSS TO REGULARLY GO .
'Carefully clean them and go to the dentist regularly.'

Figure 2: Illustration of Hindi-to-Gujarati transliteration (we refer to the result as HG), with exact matches indicated in red and partial matches in blue.

2.2.4 Finalisation of training data

The final training data for each model was the concatenation of this parallel data, the HI2GU-EN translated data and the back-translated data for that particular translation direction (See Table 4).

All synthetic data was cleaned by filtering out noisy sentences with consecutively repeated characters or tokens. As for the genuine parallel data, we choose only to use the following corpora, which contain an average sentence length of 10 tokens or more: Emille, Govin, Wikipedia and the Bible (a total of approximately 40k sentences). All data was pre-processed using FastBPE[10] with 30k BPE merge operations.

Training data source	#sents EN→GU	#sents GU→EN
Genuine parallel data	42k	42k
HI2GU-EN parallel data	1.1M	1.1M
Backtranslated monolingual	4.5M	7.1M
Total	5.6M	8.2M

Table 4: Summary of EN→GU and GU→EN training data, once filtering has been applied to synthetic data.

2.3 Supervised MT training

We trained supervised RNN (Miceli Barone et al., 2017) and transformer models (Vaswani et al., 2017) using the augmented parallel data augmented described in Section 2.2.4. For both model types, we train until convergence and then fine-tuned them on the 40k sentences of genuine parallel data, since synthetic parallel data accounted for more than 99% of total training data in both translation directions. Results are shown in Table 5, our final model results being shown in bold.

2.3.1 RNN

Our RNN submission was a BiDeep GRU sequence-to-sequence model (Miceli Barone et al., 2017) with multi-head attention.The implementation and configuration are the same as in our submission to WMT 2018 (Haddow et al., 2018), except that we use 1 attention hop with 4 attention heads, with a linear projection to dimension 256 followed by layer normalisation. Other model hyperparameters are encoder and decoder stacking depth: 2, encoder transition depth: 2, decoder base level transition depth: 4, decoder second level transition depth: 2, embedding dimension: 512, hidden state dimension: 1024. Training is performed with Adam in synchronous SGD mode with initial learning rate: 3×10^{-4}, label smoothing 0.1, attention dropout 0.1 and hidden state dropout 0.1. For the final fine-tuning on parallel data we increase the learning rate to 9×10^{-4} and hidden state dropout to 0.4 in order to reduce over-fitting.

2.3.2 Transformer

We trained **transformer base** models as defined in (Vaswani et al., 2017), consisting of 6 encoder layers, 6 decoder layers, 8 heads, with a model/embedding dimension of 512 and feed-forward network dimension of 2048.

We used synchronous SGD, a learning rate of 3×10^{-4} and a learning rate warm-up of 16,000. We used a transformer dropout of 0.1.

Our final primary systems are ensembles of four transformers, trained using different random seed initialisations. We also experimented with adjusting the weighting of the models,[11] providing gains for EN→GU but not for GU→EN, for which equal weighting provided the best results. Our final translations are produced using a beam of 12 for EN→GU and 60 for GU→EN.

2.4 Experiments and results

We report results in Table 5 on the official development set (1998 sentences) and on the official test sets (998 sentences for EN→GU and 1016 sen-

[10] github.com/glample/fastBPE.git

[11] The weights for EN→GU the were manually chosen guided by the individual BLEU scores of the models.

tences for GU→EN). Our results indicate that both the additional synthetic data as well as fine-tuning provide a significant boost in BLEU.

System	EN→GU Dev	EN→GU Test	GU→EN Dev	GU→EN Test
Semi-sup.	12.6	11.8	22.1	15.5
RNN				
+ synth. data	14.2	11.4	23.4	14.7
+ fine-tuning	15.2	11.7	24.3	15.7
Transformer				
+ synth. data	15.0	14.3	23.8	18.6
+ fine-tuning	16.9	15.1	25.9	20.6
+ Ensemble-4	17.9	16.5	27.2	**21.4**
+ Weighted Ensemble	18.1	**16.4**	-	-

Table 5: BLEU scores on the development and test sets for EN→GU. Our final submissions are marked in bold. Synthetic data is the HI2GU-EN corpus plus backtranslated data for that translation direction and fine-tuning is performed on 40k sentences of genuine parallel data.

3 Chinese ↔ English

Chinese↔English is a high resource language pair with 23.5M sentences of parallel data. The language pair also benefits from a large amount of monolingual data, although compared to English, there is relatively little in-domain (i.e. news) data for Chinese. Our aim for this year's submission was to test the use of character-based segmentation of Chinese compared to standard subword segmentation, exploiting the properties of the Chinese writing system.

3.1 Data and pre-processing

For ZH↔EN we pre-processed the parallel data, which consists of NewsCommentary v13, UN data and CWMT, as follows. The Chinese side of the original parallel data is inconsistently segmented across different corpora so in order to get a consistent segmentation, we desegmented all the Chinese data and resegmented it using the Jieba tokeniser with the default dictionary.[12] We then removed any sentences that did not contain Chinese characters on the Chinese side or contained only Chinese characters on the English side. We also cleaned up all sentences containing links, sentences longer than 50 words, as well as sentences in which the number of tokens on either side was > 1.3 times the number of tokens on the other side, following Haddow et al. (2018). After pre-processing, the corpus size was 23.6M sentences. We applied BPE with 32,000 merge operations to the English side of the corpora and then removed any tokens appearing fewer than 10 times (which were mostly noise), ending up with a vocabulary size of 32,626. For the Chinese side we attempted two different strategies: A character-level BPE model and a word-level BPE model.

Character-level Chinese A Chinese character-level model is not the same as an English character level model, as it is relatively common for Chinese characters to represent whole words by themselves (in the PKU corpus used for the 2005 Chinese segmentation bakeoff (Emerson, 2005), a Chinese word contains on average 1.6 characters). As such, a Chinese character-level model is much more similar to using a BPE model with very few merge operations on English. We hypothesised that using raw Chinese characters in tokenised text makes sense as they form natural subword units.

We segmented all Chinese sentences into characters, but kept non-Chinese characters unsegmented in order to allow for English words and numbers to be kept together as individual units. We then applied BPE with 1,000 merges, which splits the English words in the corpora into mostly trigrams and numbers as bigrams. From the resulting vocabulary we dropped characters occurring fewer than 10 times, resulting in a vocabulary of size 8,535.

We found that this segmentation strategy was successful for translating into Chinese, however produces significantly worse results when translating from Chinese into English.

Word-level Chinese For word-level Chinese, we took the traditional approach to Chinese pre-processing, where we applied BPE on top of the tokenised dataset. We used 33,000 merge operations and removed tokens occurring fewer than 10 times, resulting in a vocabulary size of 44,529.

3.2 Iterative backtranslation

We augmented our parallel data with the same backtranslated ZH↔EN as used in Sennrich et al. (2017), which consists of 8.6M sentences for EN→ZH from LDC and 9.7M sentences taken from Newscrawl for ZH→EN. After training the initial systems, we added more backtranslations for both language pairs. For the Chinese side, we used Newscrawl (2.1M sentences) as well as a re-translation of a section of LDC, ending up with

[12] https://github.com/fxsjy/jieba

9.5M sentences. For the English side we translated an additional section of Newscrawl, ending up 38M sentences in total. Much to our disappointment, we found that the extra backtranslation is not very effective at increasing the BLEU score, likely because we did not perform any specific domain adaptation for the news domain.

3.3 Architecture

We used the transformer architecture and three separate configurations.

Transformer-base This is the same architecture as described in Section 2.3.2.

Transformer-big 6 encoder layers, 6 decoder layers decoder, 16 heads, a model/embedding dimension of 1024, a feedforward network dimension of 4096 and a dropout of 0.1. For character-level Chinese, the number of layers was increased to 8 on the Chinese side. We found transformer-big to be quite fiddly to train and requires significant hyperparameter exploration. Unfortunately we were unable to find hyperparameters that work effectively for the ZH-EN direction.

Transfomer-base with larger feed-forward network We test Wang et al.'s (2018) recommendation to use the base transformer architecture and increase the feed-forward network (FFNN) size to 4096 instead of using a transformer-big model.

Ultra-large mini-batches We follow Smith et al.'s (2018) recommendation to dramatically increase the mini-batch size towards the end of training in order to improve convergence.[13] Once our model stopped improving on the development set, we increased the mini-batch size 50-fold by delaying the gradient update (Bogoychev et al., 2018) to avoid running into memory issues. This increases the average mini-batch size to 13,500 words.

3.4 Results

We identified the best single system for each language direction (Tables 6 and 7) and ensembled four models trained separately using different random seeds. We also trained right-to-left models, but they got lower scores on the development set and also did not seem to help with ensembling. Our final submission to the competition achieved 28.9 for ZH→EN and 34.4 for EN→ZH.

[13] We thank Elena Voita for alerting us to this work.

System	BLEU
Word-level segmentation for ZH	
Transformer-base	34.8
Character-level segmentation for ZH	
Transformer-base	35.1
+ Larger FFNN	35.6
Transformer-big	35.7
+ Ultra-large mini-batches	36.1

Table 6: EN→ZH results on the development set.

System	BLEU
Word-level segmentation for ZH	
Transformer-base	24.1
+ Larger FFNN	23.7
+ Ultra-large mini-batches	24.4
+ Ultra-large mini-batches	24.2
Transformer-big	11.3
Character-level segmentation for ZH	
Transformer-base	20.4

Table 7: ZH→EN results on the development set.

4 German → English

Following the success of Edunov et al. (2018) in WMT18, we decided to focus on the use of large amounts of monolingual data in the target language. In addition, we performed fine tuning on data selected specifically for the test set prior to translation, similar to the method suggested by Farajian et al. (2017), but with data selection for the entire test set instead of individual sentences.

4.1 Approach

Our approach this year is summarised as follows.

1. Back-translate all available mono-lingual English NewsCrawl data (after filtering out very long sentences). As can be seen in Table 8, the amount of monolingual data vastly outweighs the amount of parallel data available.

2. Train multiple systems with different blends of genuine parallel, out-of-domain data and back-translated in-domain data. We did not use any data from CommonCrawl or Paracrawl to train these base models.

3. For a given test set, select suitable training data from the pool of all available training data (including CommonCrawl and Paracrawl) for fine-tuning, based on n-gram overlap with the source side of the test set, focusing on rare

Corpus	Type	# of sent. pairs	# of tokens[1] (DE)	# of tokens (EN)
Europarl v9	parallel	1.82 M	48.66 M	51.15 M
Rapid 2019	parallel	1.48 M	30.56 M	30.95 M
News Commentary	parallel	0.33 M	8.51 M	8.51 M
CommonCrawl[1]				
as distributed	parallel	2.40 M	56.87 M	60.83 M
filtered	parallel	0.87 M	19.54 M	20.23 M
ParaCrawl v3[2]				
as distributed	parallel	31.36 M	596.66 M	630.50 M
filtered	parallel	16.66 M	328.14 M	343.68 M
News Crawl 2007–2018	English[3]	199.74 M	4,764.26 M	4,805.45 M

[1] continuous sequences of letters, digits, or repetitions of the same symbol; otherwise, a single symbol.
[2] used for fine-tuning but not for training the base models, filtered as described in Section 4.4.
[3] German side obtained by back-translation with a model from our participation in WMT18.

Table 8: Training data used for German→English translation.

n-grams that occur fewer than 50 times in the respective sub-corpus[14] of training data.

4. Finally, we translate with an ensemble over several check-points of the same training run (best BLEU prior to fine-tuning, fine-tuned, best mean cross-entropy per word if different from best BLEU, etc.).

4.2 Data Preparation

4.2.1 Tokenisation Scheme

For tokenisation and sub-word segmentation, we used SentencePiece[15] (Kudo and Richardson, 2018) with the BPE segmentation scheme and a joint vocabulary of 32,000 items.

4.3 Back-translation

We back-translated all of the available English NewsCrawl data using one of the models from our participation in the WMT18 shared task.

4.4 Data Filtering

The CommonCrawl and ParaCrawl datasets consist of parallel data automatically extracted from web pages from systematic internet crawls. These datasets contain considerable amounts of noise and poor quality data. We used dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018) to rank the data in terms of estimated translation quality, and only retained data that scored higher than a threshold determined by cursory inspection of the data by a competent bilingual at various threshold levels. Table 8 shows the amounts of raw and filtered data. For training, we limited the training data to sentence pairs of at most 120 SentencePiece tokens on either side (source or target).

4.5 Model Training

4.5.1 Initial Training

To investigate the effect of the blend of genuine parallel and back-translated news data on translation quality, we trained five transformer-big models (cf. Section 3.3) with different blends of back-translated and genuine parallel data.

We used a dropout value of 0.1 between transformer layers and no dropout for attention and transformer filters. We used the Adam optimiser with a learning rate of 0.0002 and linear warm-up for the first 8K updates, followed by inverted squared decay.

Figure 3 shows the learning curves for these five initial training runs as validated against the WMT18 test set. Note that the BLEU scores are inflated, as they were computed on the sub-word units rather than on de-tokenised output. The curves suggest that adding large amounts of training data does improve translation quality in direct comparison between the different training runs. However, compared to last year's top system submissions, these systems were still lagging behind.

4.5.2 Continued training with increased batch size

Similar to our EN↔ZH experiments, we experiment with drastically increasing the mini-batch size by increasing optimiser delay (cf. Section 3.3). Figure 4 shows the effect of increased mini-batch sizes of ca. 9K, 13K, and 22K sentence pairs, re-

[14] For practical reasons, we sharded the training data based on provenance. In addition, each year of the backtranslated news data was treated as a separate sub-corpus.
[15] https://github.com/google/sentencepiece

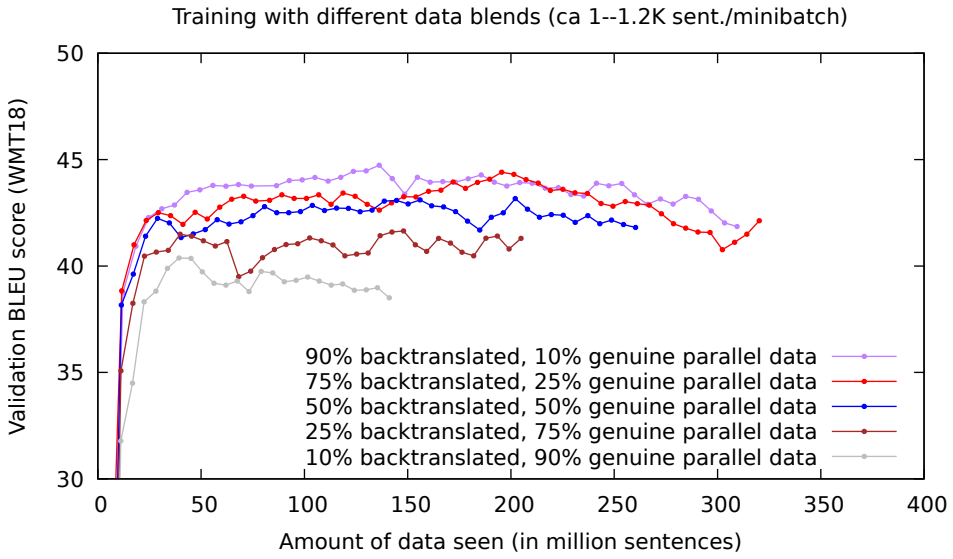

Figure 3: Learning curve for different blends of genuine parallel and synthetic back-translated data. Note that the BLEU scores are inflated with respect to SACREBLEU as they are calculated on BPE-segmented data.

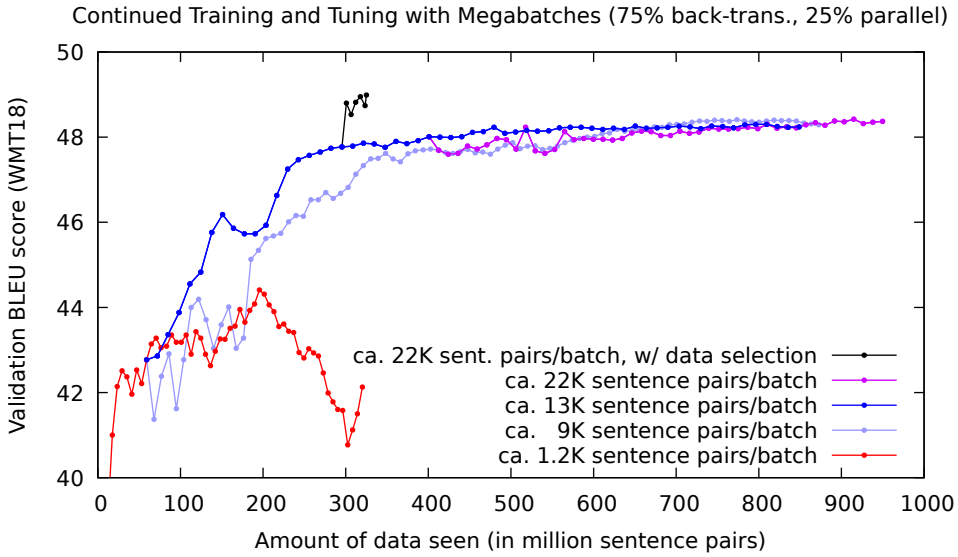

Figure 4: Effect of increased batch size for training and of tuning on data selected for the test set. The red line shows the learning curve for the original training settings (mini-batches of ca. 1,200 sentence pairs). The other lines are the learning curves for models that were initialised with the model parameters of another model at some point in its training process (specifically: at the point where the new learning curve branches off), and then trained with increased batch sizes on the same data (blue and magenta lines), or on data specifically selected to contain rare n-grams that also occur in the test / validation set.

spectively. The plot shows drastic improvements in the validation scores achieved.

4.5.3 Fine-tuning on selected data

As a last step, we selected data specifically for the test set and continued training on this data for one epoch of this data. For the WMT18 test set, this gives a significant boost over the starting point, as the black line in Figure 4 shows.

4.6 Results and Analysis

Due to resource congestion, we were not able to train our models to convergence in time for submission. The point where the black line in Figure 4 branches off shows the state of our models prior to tuning for a specific test set.

For our submission to the shared task, we ensembled four models:

- an untuned model trained on a blend of 75% back-translated data and 25% genuine parallel data
- checkpoint models after 500, 2000, and 3000 updates with batches of ca. 13K sentences on data selected specifically for the WMT19 test set. This data included data from CommonCrawl and Paracrawl.

With a BLEU score of 36.7 (35.0 cased) — as opposed to 44.3 (42.8 cased) for the top-performing system — our results were disappointing. Apart from a probably suboptimal choice of training hyperparameters, what else went wrong?

Post-submission analysis In order to understand the effect of back-translations better, we evaluated our systems on a split of test sets from past years into "forward" (German is the original source language) and "reverse" (the source side of the test set are German translations of texts originally written in English). The results are shown in Table 9. As we can see, most of the gains from using back-translations are concentrated in the "reverse" section of the test sets. The same also holds for Edunov et al.'s (2018) results on the WMT18 test sets for en→de. Notice how it outperforms the top-performing system (Microsoft Marian) on the reverse translation direction but lags behind in the forward translation.[16]

[16]We thank Barry Haddow for pointing this out to us and for providing us with the split test sets and the split numbers for the Microsoft and Facebook systems.

We see two possible reasons for this phenomenon. The first is that back-translations produce synthetic data that is closer to the reverse scenario: translating back from the translation into the source. The second reason is that the reverse scenario offers a better domain match: newspapers tend to report relatively more on events and issues relating to their local audience. A newspaper in Munich will report on matters relating to Munich; the Los Angeles time will focus on matters of interest to people living in Southern California.

This became evident when we investigated some strange translation errors that we observed in our submission to the shared task. For example, our system often translates "Münchnerin" (woman from Munich) as 'miner', 'minder', or 'mint' and "Schrebergarten" (allotment garden) as 'shrine' (German: Schrein). When we checked our back-translated training data for evidence, we noticed that these are systematic translation errors in our back-translations. While the word "Münchnerin" is frequent in our German data, women from Munich are rarely mentioned as such in English newspapers. With BPE breaking up rare words into smaller units, the system learned to translate "min" (possibly from "min|t" (as in the production facility for coins), which is "Mün|ze" or "Mün|zprägeanstalt" in German) into "Mün". Once "Mün" was chosen in the decoder of the MT system, the German language model favored the sequence *Mün|ch|nerin* over *Mün|ze* or the even rarer *Münzprägeanstalt*.

These findings suggest that back-translated data as well needs curation for domain match and systematic translation errors.

Since this year's test sets consist only of the (more realistic) "forward" scenario, we were not able to replicate the gains we observed for previous test sets when adding more back-translated data.

5 English → Czech

English-Czech is a high-resource language pair in the WMT News Translation shared task. For our submission to the EN→CS track, we investigated the effects of simplifying the data pre-processing and training data filtering, and experimented with larger architectures of the Transformer model.

5.1 Data and pre-processing

For English→Czech experiments we use all parallel corpora available to build a constrained system except CommonCrawl, which is noisy and rela-

System	batch[1]	WMT15 fwd	WMT15 rev	WMT16 fwd	WMT16 rev	WMT17 fwd	WMT17 rev	WMT18 fwd	WMT18 rev
10% back-translated, 90% parallel	1.2K	20.4	34.9	27.7	44.4	25.1	37.8	28.5	46.7
25% back-translated, 75% parallel	1.2K	20.0	37.7	27.5	47.5	24.9	39.8	27.5	49.4
50% back-translated, 50% parallel	1.2K	20.2	38.3	28.2	48.8	25.9	40.8	28.3	51.3
75% back-translated, 25% parallel	1.2K	20.9	39.0	29.4	49.7	26.6	41.7	29.6	52.4
90% back-translated, 10% parallel	1.2K	21.2	38.6	29.0	49.6	26.8	41.5	29.7	52.8
75% back-translated, 25% parallel	1.2K	20.9	39.0	29.4	49.7	26.6	41.7	29.6	52.4
75% back-translated, 25% parallel	9K	23.2	41.2	31.8	51.8	28.7	44.2	32.6	56.3
75% back-translated, 25% parallel	13K	23.2	40.9	31.8	51.3	28.6	44.1	32.4	56.2
75% back-translated, 25% parallel	22K	23.2	41.2	31.8	51.3	28.7	44.2	32.4	56.2
75/25, with tuning for WMT18	22K	23.6	41.3	32.5	51.6	28.9	44.0	33.2	56.7
							Microsoft Marian 2018 (en→de)	52.5	41.6
							Edunov et al. (2018) (en→de)	45.8	46.1

[1] batch size in sentence pairs

Table 9: Contrastive evaluation (BLEU scores) of performance on genuine German → English (fwd) translation vs. English source restoration from text originally translated from English into German (rev).

tively small compared to the CzEng 1.7 corpus[17] (Bojar et al., 2016). We clean the data following Popel (2018) by removing sentence pairs that do not contain at least one Czech diacritic letter. Duplicated sentences, sentences with <3 or >200 tokens, and sentences with the ratio of alphabetic to non-alphabetic characters <0.5 are also removed. The final parallel training data contains 44.93M sentences. For back-translation we use approximately 80M English and Czech monolingual sentences from NewsCrawl (Bojar et al., 2018), which we cleaned in a similar manner.

Preprocessing	Dev	2017	2018
Tc + Tok + BPE	26.8	23.0	22.2
Tc + Tok + ULM	26.7	22.9	22.3
ULM (raw text)	26.7	22.9	**22.9**
+ Resampling	26.7	22.2	21.8

Table 10: Comparison of different pre-processing pipelines for EN→CS according to BLEU. *Tc* stands for truecasing, *Tok* for tokenisation.

We aimed to explore whether, in a high-resource setting, the common pre- and post-processing pipelines that usually include truecasing, tokenisation and subword segmentation using byte pair encoding (BPE) (Sennrich et al., 2016b) can be simplified with no loss to performance. We replace BPE with the segmentation algorithm based on a Unigram Language Model (ULM) from SentencePiece, which is built into Marian. In both cases we learn 32k subword units jointly on 10M sampled English and Czech sentences. We gradually remove the elements of the pipeline and find no significant difference between the two segmentation algorithms (Table 10). We do observe a performance drop when subword resampling is used, but this has been shown to be more effective particularly for Asian languages (Kudo, 2018). For the following English-Czech experiments, we use ULM segmentation on raw text.

[17] https://ufal.mff.cuni.cz/czeng/czeng17

5.2 Experiment settings

We use the transformer-base and transformer-big architectures described in Section 3.3. Models are regularised with dropout between transformer layers of 0.2 and in attention of 0.1 and feed-forward layers of 0.1, label smoothing and exponential smoothing: 0.1 and 0.0001 respectively. We optimise with Adam with a learning rate of 0.0003 and linear warm-up for first 16k updates, followed by inverted squared decay. For Transformer Big models we decrease the learning rate to 0.0002. We use mini-batches dynamically fitted into 48GB of GPU memory on 4 GPUs and delay gradient updates to every second iteration, which results in mini-batches of 1-1.2k sentences. We use early stopping with a patience of 5 based on the word-level cross-entropy on the *newsdev2016* data set. Each model is validated every 5k updates, and we use the best model checkpoint according to uncased BLEU score.

Decoding is performed with beam search with a beam size of 6 with length normalisation. Additionally, we reconstruct Czech quotation marks using regular expressions as the only post-processing step (Popel, 2018).

5.3 Experiments and Results

Lang.	System	Dev	2017	2018
EN-CS	Transformer-base	26.7	22.9	22.9
	+ Data filtering	27.1	23.4	22.6
CS-EN	Transformer-base	32.6	28.8	30.3
	+ Back-translation	37.3	31.9	32.4
EN-CS	Base + Back-transl.	28.4	25.1	25.1
	→ Transformer-big	29.6	26.3	26.2
	+ Ensemble x2	29.6	26.5	26.3

Table 11: BLEU score results for EN-CS experiments.

Results of our models are shown in Table 11.

We first trained single transformer-base models for each language direction to serve as our baselines. We then re-score the EN→CS training data using the CS→EN model and filter out the 5% of data with the worst cross-entropy scores, which is a one-directional version of the dual conditional cross-entropy filtering, which we also used for our EN→DE experiments. This improves the BLEU scores on the development set and *newstest2017*. Next, we back-translate English monolingual data and train a CS→EN model, which in turn is used to generate back-translations for our final systems. The addition of back-translated data improves the Transformer Base model by 1.7-2.5 BLEU, which is less than the improvement from iterative back-translations reported by (Popel, 2018). A Transformer Big model trained on the same data is ca. 1.1 BLEU better.

Due to time and resource constraints we train and submit a EN→CS system (this was the only language direction for English-Czech this year) consisting of just two transformer-big models trained with back-translated data. Our system achieves 28.3 BLEU on *newstest2019*, 2.1 BLEU less then the top system, which ranks it in third position.

6 Summary

This paper reports the experiments run in developing the six systems submitted by the University Edinburgh to the 2019 WMT news translation shared task. Our main contributions have been in different exploitation of additional non-parallel resources, in investigating different pre-processing strategies and in the testing of a variety of NMT training techniques. We have shown the value of using additional monolingual resources through pre-training and semi-supervised MT for our low-resource language pair EN-GU. For the higher resource language pairs, we also exploit monolingual resources in the form of backtranslation. For GU→EN in particular we study the effect on translation quality of varying the ratio between between genuine and synthetic parallel training data. For EN→ZH, we showed that character-based decoding into Chinese produces better results than the standard subword segmentation approach. In EN→CS, we also studied the effects of pre-processing, by showing that in such a high resource setting, a simplified pre-processing pipeline can be highly successful.

Our low resource language pairs, EN→GU and GU→EN systems were ranked 1st and 2nd respectively out of the constrained systems according to the automatic evaluation. For the high resource pairs, our EN→CS system ranked 3rd, EN→ZH and ZH→EN ranked 7th and 6th respectively and DE→EN ranked 9th.

Acknowledgements

This work was supported by funding from the European Union's Horizon 2020 research and innovation programme under grant agreements No 825299 (GoURMET), 825303 (Bergamot), and 825627 (European Language Grid).

It was also supported by the UK Engineering and Physical Sciences Research Council (EPSRC) fellowship grant EP/S001271/1 (MTStretch).

It was performed using resources provided by the Cambridge Service for Data Driven Discovery (CSD3) operated by the University of Cambridge Research Computing Service (http://www.csd3.cam.ac.uk/), provided by Dell EMC and Intel using Tier-2 funding from the Engineering and Physical Sciences Research Council (capital grant EP/P020259/1), and DiRAC funding from the Science and Technology Facilities Council (www.dirac.ac.uk).

We express our warmest thanks to Kenneth Heafield, who provided us with access to the computing resources.

References

Nikolay Bogoychev, Marcin Junczys-Dowmunt, Kenneth Heafield, and Alham Fikri Aji. 2018. Accelerating asynchronous stochastic gradient descent for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP'18, pages 2991–2996, Brussels, Belgium.

Ondřej Bojar, Ondřej Dušek, Tom Kocmi, Jindřich Libovický, Michal Novák, Martin Popel, Roman Sudarikov, and Dušan Variš. 2016. CzEng 1.6: Enlarged Czech-English Parallel Corpus with Processing Tools Dockered. In *Text, Speech, and Dialogue: 19th International Conference, TSD 2016*, number 9924 in Lecture Notes in Computer Science, pages 231–238, Brno, Czech Republic.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, and Christof Monz. 2018. Findings of the 2018 Conference on Machine Translation (WMT18). In *Proceedings of the 3rd Conference on Machine Translation, Volume 2: Shared Task Papers*, WMT'18, pages 272–307, Belgium, Brussels.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics; Human Language Technologies*, NAACL-HLT'19, Minneapolis, Minnesota.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding Back-Translation at Scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, EMNLP'18, pages 489–500, Brussels, Belgium.

Thomas Emerson. 2005. The Second International Chinese Word Segmentation Bakeoff. In *Proceedings of the 4th SIGHAN Workshop on Chinese Language Processing*, Jeju Island, Korea.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the 2nd Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Advances in Neural Information Processing Systems*, pages 1019–1027.

Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann, Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone, and Rico Sennrich. 2018. The University of Edinburgh's Submissions to the WMT18 News Translation Task. In *Proceedings of the 3rd Conference on Machine Translation*, WMT'18, pages 399–409, Brussels, Belgium.

Cong Duy Vu Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative Back-Translation for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, WNMT'18, pages 18–24, Melbourne, Australia.

Marcin Junczys-Dowmunt. 2018. Microsoft's submission to the WMT2018 news translation task: How I learned to stop worrying and love the data. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 425–430, Belgium, Brussels. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast Neural Machine Translation in C++. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, ACL'18, pages 116–121, Melbourne, Australia.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, ICLR'15, San Diego, California, USA.

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 45th Annual Meeting of the Association for Computational Linguistics*, ACL'07, pages 177–180, Prague, Czech Republic.

Taku Kudo. 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 (Volume 1: Long Papers)*, pages 66–75, Melbourne, Australia. Association for Computational Linguistics.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual Language Model Pretraining. In *arXiv:1901.07291*.

Antonio Valerio Miceli Barone, Jindřich Helcl, Rico Sennrich, Barry Haddow, and Alexandra Birch. 2017. Deep Architectures for Neural Machine Translation. In *Proceedings of the 2nd Conference on Machine Translation, Volume 1: Research Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Martin Popel. 2018. CUNI transformer neural mt system for WMT18. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 486–491, Belgium, Brussels. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the 3rd Conference on Machine Translation: Research Papers*, pages 186–191, Belgium, Brussels.

Rico Sennrich, 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 2nd Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 389–399, Copenhagen, Denmark.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16, pages 86–96, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, ACL'16, pages 1715–1725, Berlin, Germany.

Samuel L. Smith, Pieter-Jan Kindermans, Chris Ying, and Quoc V. Le. 2018. Don't decay the learning rate, increase the batch size. In *Proceedings of the 6th International Conference on Learning Representations*, ICLR'18, Vancouver, Canada.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Qiang Wang, Bei Li, Jiqiang Liu, Bojian Jiang, Zheyang Zhang, Yinqiao Li, Ye Lin, Tong Xiao, and Jingbo Zhu. 2018. The NiuTrans Machine Translation System for WMT18. In *Proceedings of the 3rd Conference on Machine Translation: Shared Task Papers*, WMT'18, pages 528–534, Belgium, Brussels.