The University of Edinburgh's Submissions to the WMT18 News Translation Task

Barry Haddow, Nikolay Bogoychev, Denis Emelin, Ulrich Germann,
Roman Grundkiewicz, Kenneth Heafield, Antonio Valerio Miceli Barone and **Rico Sennrich**
University of Edinburgh, Scotland

Abstract

The University of Edinburgh made submissions to all 14 language pairs in the news translation task, with strong performances in most pairs. We introduce new RNN-variant, mixed RNN/Transformer ensembles, data selection and weighting, and extensions to back-translation.

1 Introduction

For the WMT18 news translation task, we were the only team to make submissions to all 14 language pairs. Our submissions built on our strong results of the WMT16 and WMT17 tasks (Sennrich et al., 2016a, 2017), in that we used neural machine translation (NMT) with byte-pair encoding (BPE) (Sennrich et al., 2016c), back-translation (Sennrich et al., 2016b) and deep RNNs (Miceli Barone et al., 2017). For this year's submissions we experimented with new architectures, and new ways of data handling. In brief, the innovations that we introduced this year are:

Architecture This year we experimented with the Transformer architecture (Vaswani et al., 2017), as implemented by Marian (Junczys-Dowmunt et al., 2018), as well as introducing a new variant on the deep RNN architectire (Section 2.3).

Data selection and weighting For some language pairs, we experimented with different data selection schemes, motivated by the introduction of the noisy ParaCrawl corpora to the task (Section 2.1). We also applied weighting of different corpora to most language pairs, particularly DE↔EN (Section 3.5).

Extensions to Back-translation For TR↔EN (Section 3.7) we used copied monolingual data (Currey et al., 2017a) and iterative back-translation.

In-domain Fine-tuning For RU↔EN (Section 3.6) we fine-tuned using a specially constructed "in-domain" data set.

2 System Details

In this section we describe the general properties of our systems, as well as some novel approaches that we tried this year such as data selection and a variant on the GRU-based RNN architecture. The specifics of our submissions for each language pair are described in Section 3.

2.1 Data and Selection

All our systems were constrained in the sense that they only used the supplied parallel data (including ParaCrawl) for training the systems. We also used the monolingual news crawls to create extra synthetic parallel data by back-translation, for all language pairs, and by copying monolingual data for TR↔EN. During training we generally used *newsdev2016* or *newstest2016* for validation, and *newstest2017* for development testing (i.e. model selection), except for ZH↔EN, and ET↔EN, where we used the recent *newsdev* sets instead.

All parallel data contains a certain amount of noise, and the problem was exacerbated this year since the organisers provided a ParaCrawl corpus[1] for most language pairs[2] as additional training data. On inspection, we could see that these crawled corpora were quite noisy, including mis-aligned sentence pairs, incorrect language, and garbled encodings. In early experiments, we showed increases in BLEU from including ParaCrawl in the training data, for ET→EN and FI→EN, but we decided to see if we could improve performance further by applying data filtering. We experimented with different filtering methods, described below.

[1] `https://paracrawl.eu`

[2] ParaCrawl corpora was not available for EN↔TR and EN↔ZH.

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 399–409
Belgium, Brussels, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64039

Language Identifier Filtering This was applied to the CS↔EN and DE↔EN corpora, based on observations that CzEng, and ParaCrawl both contain sentence pairs in the "wrong" language. For CS↔EN we applied langid (Lui and Baldwin, 2012) to both sids of the data, removing any sentences whose English side is not labelled as English, or whose Czech is not labelled as Czech, Slovak or Slovenian[3]. For DE↔EN, we just applied langid to ParaCrawl and retained only those pairs where each side was identified as the 'correct' language by *langid*. This reduced the size of the ParaCrawl corpus from about 36 million sentence pairs to ca. 18 million sentence pairs.

Data Selection with Translation Perplexity We applied this to ET↔EN and FI↔EN. To perform the filtering, we first trained shallow RNN models in both directions, using all the permitted parallel data except ParaCrawl. We then used these models to score the ParaCrawl sentence pairs, normalising by target sentence length, and adding the scores for forward and reverse models. We then ranked sentence pairs in ParaCrawl using this score, and performed a grid search across different thresholds (from 0 – 100% in 10 point intervals) of the ParaCrawl data, in addition to the other parallel data. We trained a shallow RNN system using the data selected across each of these thresholds, and tested it on *newstest2017* (for FI→EN), or half of *newsdev2018* (for ET→EN).

The results of the filtering are shown in Figure 1. Based on these results, we chose a threshold of 0.3 for ET↔EN (which gives us +0.8 BLEU), but used the whole of ParaCrawl for FI→EN.

Alignment-based Filtering We applied this to the DE→EN parallel data, after langid filtering. We word-aligned all pre-cleaned parallel data with *fastalign* (Dyer et al., 2013) and computed the geometric mean of forward and backward alignment probabilities as a coarse estimate of how good a translation pair the respective sentence pair is.

All parallel data was sorted in descending order of this "plausible translation" score, and a neural system was trained on this data, in this order. In order to determine a threshold for data filtering, we monitored the performance on a validation set (*newstest2016*) and observed the point where translation quality started to deteriorate. We used the translation plausibility score at this point as the threshold for selecting data for training the final systems.

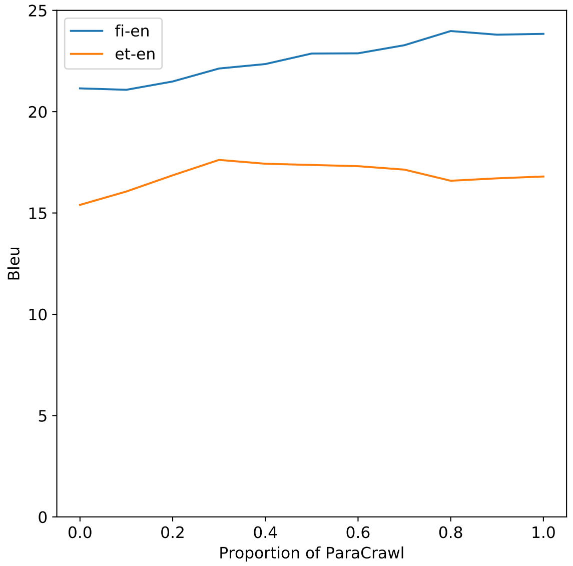

Figure 1: Result of translation perplexity filtering of ParaCrawl on 2 language pairs

2.2 Preprocessing

For most language pairs, our preprocessing setup consisted of the Moses pipeline (Koehn et al., 2007) of normalisation, tokenisation and truecasing, followed by byte-pair encoding (BPE) (Sennrich et al., 2016c). We generally applied joint BPE, with the number of merge operations set on a per-pair basis, detailed in Section 3. Different pipelines were used for processing the two languages written in non-Latin scripts (i.e. Chinese and Russian), also explained in Section 3. For some language pairs (those including Czech, Estonian, Finnish and German) we used the preprocessed data provided by the organisers (which is preprocessed up to truecasing), whilst for the others we started with the raw data.

2.3 Model Architecture

For this submission we considered two types of sequence-to-sequence architectures: a transformer (Vaswani et al., 2017) and a deep RNN, specifically the BiDeep GRU encoder-decoder (Miceli Bar-

[3] langid identified a significant proportion of the data as these other two Slavic languages, but on inspecting a sample, they were found nearly always to be Czech. The issue with langid is that we just give it the text, without providing any prior knowledge, when in actual fact there is a strong prior that CzEng sentences are really Czech and English, by construction

one et al., 2017). Both architectures[4] are implemented in the Marian open source neural machine translation framework (Junczys-Dowmunt et al., 2018). For the transformer architecture we used the "wmt2017-transformer" setup from the Marian example collection[5].

We extended the RNN with with multi-head and multi-hop attention. Multi-head attention is similar to Chen et al. (2018), with an MLP attention mechanism using a single $\tanh$ hidden layer followed by one soft-max layer for each attention heads. We further include an optional projection layer on the attended context with layer normalisation in order to avoid increasing the total size of the attended context.

Let $C \in \mathcal{R}^{N_s \times d_e}$ be the input sentence representation produced by the encoder, where N_s is the source sentence length and d_e is the top-level bidirectional encoder state dimension. Let $s \in \mathcal{R}^{d_d}$ be an internal decoder state at some step. Then for source sentence position i we compute a vector of M attention weights, where M is the number of attention heads:

$$\begin{aligned} W, A &\in \mathcal{R}^{N_s \times M} \\ W_i &= \text{MLP}(C_i, s) \\ A_i &= \frac{\exp(W_i)}{\sum_{i'} \exp(W_{i'})} \end{aligned}$$

where we assume that exponentiation is applied element-wise. Then we compute the attended context vector as:

$$\text{ATT}(C, h) = \text{CAT}_{r=1}^{M} \left(\sum_i \text{PROJ}_r(C_i) \cdot a_{i,r} \right)$$

where $CAT_{r=1}^{M}$ is vector concatenation over the attention heads and each $PROJ_r$ is either the identity function or a trainable linear layer followed by layer normalization.

Multi-hop attention is similar to Gehring et al. (2017), except that we do not use convolutional layers, but instead we introduce additional attention hops between the layers of the deep transition GRU in the decoder. In our implementation multi-head and multi-hop attention can be combined, in which case each attention hop is a separate multi-head attention mechanism.

Let $L_t \geq 2$ be the decoder base recurrence depth and $H < L_t$ be the number of attention hops. Then the base level of the decoder is defined as:

$$\begin{aligned} s_{j,1} =& \text{GRU}_1 \left(y_{j-1}, s_{j-1,L_t} \right) \\ s_{j,k} =& \text{GRU}_k \left(\text{ATT}_k(C, s_{j,k-1}), s_{j,k-1} \right) \\ & \text{for } 1 < k \leq H + 1 \\ s_{j,k} =& \text{GRU}_k \left(0, s_{j,k-1} \right) \\ & \text{for } H + 1 < k \leq L_t \end{aligned}$$

where each $\text{ATT}_k(C, s)$ is and independent multi-head attention mechanism with M heads. For a BiDeep decoder, the higher levels are the same as in the default Marian implementation of the BiDeep architecture [6].

2.4 Training

All our systems are trained with Marian[7] (Junczys-Dowmunt et al., 2018), using Adam (Kingma and Ba, 2015). To improve training stability and generalisation, we employed label smoothing (0.1) (Szegedy et al., 2016), exponential smoothing (i.e. Polyak averaging) with 0.0001 weight, gradient clipping and layer normalisation (Ba et al., 2016). For all pairs except CS↔EN (where it harmed BLEU) we used dropout (Srivastava et al., 2014; Gal and Ghahramani, 2016) on the Transformer/RNN connections.

3 Submitted Systems

3.1 Chinese ↔ English

For ZH↔EN we preprocessed the parallel data, which consists of NewsCommentary v13, UN data and CWMT, as follows. We first desegmented all the Chinese data and resegmented it using Jieba[8]. 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 where the amount of tokens on either side was $>$ 1.3 times the tokens on the other side, following Hassan et al. (2018). After preprocessing the corpus size was 23.6M sentences. We then applied BPE using 18,000 merge operations and we used the top 18,000 BPE segments as vocabulary. We augmented our data with backtranslated

[4] The BiDeep GRU is obtainable using the `-best-deep` option.

[5] https://github.com/marian-nmt/marian-examples

[6] The implementation of the multi-head and multi-hop attention architectures is available at: https://github.com/EdinburghNLP/marian-dev

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

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

ZH↔EN from Sennrich et al. (2017), which consists of 8.6M sentences for EN→ZH and 19.7M for ZH→EN.

We trained using the BiDeep architecture with multi-head attention with 1 hop and 3 heads. We decoded using an ensemble of 5 L2R systems and a beam of 12 for EN→ZH and 6 L2R systems and a beam of 12 for ZH→EN. Due to time constraints, we were not able to train any of the systems to convergence.

3.2 Czech ↔ English

After preprocessing, language filtering (see Sections 2.1 and 2.2), and removing any parallel sentences where neither side contains an ASCII letter, we were left with around 50M sentence pairs. We then learned a joint BPE model over the source and target corpora, with 89,500 merge operations, and applied it using a vocabulary threshold of 50.

For back-translation, we trained shallow RNN models in both directions without ParaCrawl or the langid-based corpus cleaning, and used to decode with a beam size of 5. We back-translated the English 2017 news-crawl, and the Czech news-crawls from 2016 and 2017, removing lines with more than 50 tokens, to create additional corpora of approximately 26.5M sentence for CS→EN and 13M for EN→CS. Initially we tried simply concatenating each of these corpora with the natural parallel data, but this gave poor results for CS→EN, so we over-sampled the synthetic data 2 times for that pair to give approximately equal amounts for synthetic and natural data. For EN→CS, we did not see any benefit from equalising the synthetic/natural ratio, so we stuck to using simple concatenation.

For the submitted systems, we trained the BiDeep RNN models using Marian. In addition to the default Marian settings, we used layer normalisation, tied embeddings, label smoothing (0.1), exponential smoothing, no dropout, but we used multiheaded/multihop attention with 2 heads and 3 hops. We trained on 4 GPUs with a working memory of 4000MB on each, validating every 2,500 updates. We used exponential smoothing and took the final smoothed model. We trained 4 left-right (L2R) and 4 right-left (R2L) models for each language pair, and due to time constraints we did not train to convergence, stopping each run after about 250k–350k updates. We decoded using an ensemble of the 4 L2R systems and a beam of 50, then reranked with the 4 R2L systems. For both language pairs we normalised probabilities by target length, raising it to a power of 0.8 for CS→EN.

3.3 Estonian ↔ English

As explained in Section 2.1, we used a filtered ParaCrawl for this pair, and in common with CS↔EN we removed any sentence pairs where either side contained no ascii letter. We trained and applied a BPE model with 89,500 merge operations and a vocabulary threshold of 50. We split *newsdev2018* randomly and used one half for validation and another half for development testing.

The models used for back-translation were shallow RNNs trained on the parallel data without ParaCrawl. We translated the 2017 English news-crawl to Estonian, and translated all the Estonian news-crawls to English. We also experimented with the BigEst Estonian corpus, but did not see any improvement when using it to produce synthetic data, nor when we selected 50% of it using Moore-Lewis selection (Moore and Lewis, 2010) with the news-crawl data as in-domain. Our final natural parallel corpus contains approximately 1.2M sentences, and the synthetic corpora are about 2.9M for EN→ET and 26.5M for ET→EN. To create the final corpora for training, we combined natural and synthetic, over-sampling the natural 3-times for EN→ET and 23-times for the ET→EN. Again we apply BPE, trained on the Europarl, Rapid and selected Paracrawl corpora, with the same parameters as before.

Our submitted system was an ensemble of 4 left-right systems, reranked with 4 right-left systems, with each ensemble consisting of 2 deep BiDeep RNNs and 2 Transformers. The RNN had a BiDeep architecture, with layer normalisation, tied embeddings, label smoothing (0.1), exponential smoothing, RNN dropout (0.2), source and target word dropout (0.1) and multihead/multihop attention with 2 heads and 3 hops. We trained on 4 GPUs with a working memory of 4000MB on each, validating every 2,500 updates. The RNNs were not trained to convergence (due to time constraints) but stopped after between 300k and 500k steps. The transformer models used the settings from Marian examples. without layer normalisation, with a working memory of 9500MB (on each of 4 GPUs), validating every 2500 updates, and detecting convergence with a patience of 10. We also applied source and target word dropout to the transformer models. They generally converged in under 200k updates. As for CS↔EN we used exponentially smoothed models. Decoding is the same as for CS↔EN, with normalisation by target length.

3.4 Finnish ↔ English

For FI↔EN, after pre-processing we removed sentence pairs where either side contains no ascii characters, then trained and applied a BPE model with 89,500 merge operations and a vocabulary threshold of 50. As reported in Section 2.1, we used the whole of ParaCrawl in our system.

For back-translation, we trained shallow RNN models in each direction, without ParaCrawl. We back-translated with a beam size of 5, translating the English 2017 news-crawl to Finnish, and the Finnish 2014–2017 news-crawls to English. Before back-translation, we removed any sentences with length greater than 50 tokens. For EN→FI, we combined 3.2M naturally parallel sentence pairs, over-sampling 5-times, with 14.6M sentences of synthetic data. For FI→EN, we combined the same natural corpus (over-sampled 8-times) with a 26.5M corpus of synthetic parallel data.

We created the submitted systems in the same way as the ET↔EN systems (Section 3.3), and again we were not able to train the deep RNNs to convergence. The only difference is that for EN→FI, we normalise by the target length raised to a power of 0.5, after running a grid search over different normalisations on the development set.

3.5 German ↔ English

Our efforts focussed on extracting the most useful data from ParaCrawl. After preprocessing and selection (see Section 2.1, we trained and applied joint BPE models with 35,000 merge operations, and a threshold of 50.

To balance the data, we blended the data in a mix as shown in Table 1, by randomly sampling from each corpus (without raplacement), resetting (i.e., replacing all items at once) each corpus when it became exhausted, for a total of 40 million sentence pairs.

Our system was based on the transformer in Marian examples, and initially we trained several left-right and right-left systems with tied target embeddings (but separate source embeddings). We used these systems to create ensembles.

For the translation direction EN→DE, we also trained a single model with a set-up more closely reflecting the setup described in the *wmt2017-transformer* Marian example set-up. For this single decoder, we tied all embeddings and pooled the top-ranked 7.5 million sentence pairs from paracrawl (according to the translation plausibility score) with the other training data. Below, this system is referred to as *single transformer*.

Corpus	%
Back translations[1]	50%
CommonCrawl	5%
Europarl	15%
News-commentary	10%
ParaCrawl	10%
Rapid	10%

Table 1: Blend of data for training the DE↔EN ensemble models (40M sentence pairs total).

For the single transformer we used a mix of approximately 4.6 million parallel sentence pairs from latest versions of Europarl, CommonCrawl and News-commentary, oversampled twice, the 7.5 million parallel sentence pairs from ParaCrawl, filtered as described above, and 10 million back-translated sentences from NewsCrawl 2016. We trained a Marian transformer model with standard settings.

We also ran preliminary experiments with multi-head and multi-hop GRU architectures on the same training data except ParaCrawl but we found that these models tended to underperform the transformer by $0.6 - 1.0$ BLEU points, therefore we did not use them for our submission.

As the results in Table 2 show, the single transformer produces better results than our ensembles. Even re-ranking of the single transformer output deteriorates the results, which we attribute to lower quality of the models used for ensembling and re-ranking. At this point we do not know whether the differences in model quality are due to differences in the tying of parameters, different choices of other hyperparameters, differences in the training data used, or a combination of any of these potential causes.

3.6 Russian ↔ English

After preprocessing, we trained a joint BPE model with 90,000 merge operations, using the same Latin-Cyrillic transliteration trick as in Sennrich et al. (2016c). For back-translation we trained a deep RNN and translated Russian news crawls

Pair	System	BLEU
DE→EN	Ensemble of 3 L2R, reranked with ensemble of 2 R2L	43.9
EN→DE	Single transformer	44.4
	Single transformer, reranked with ensemble of 2 R2L	43.2
	Ensemble of 2 L2R, reranked with ensemble of 2 R2L	41.8

Table 2: WMT18 Results for German ↔ English

from 2015–2017, and the English news crawl from 2017 to give about 36M sentences in each direction.

In order to maximize the performance of our submission systems, we created a pseudo "in-domain" fine-tuning corpus designed to be representative of the targeted news domain to a greater extent than the full parallel corpus. For that purpose, we concatenated pre-processed sentence pairs from NewsCommentary v13, CommonCrawl, and Yandex Corpus, excluding the noisy ParaCrawl data as well as data from the UN Parallel Corpus V1.0 which has little overlap with our target domain. To ensure that the so assembled corpus is as free of noise as possible, we furthermore filter out sentence pairs in which the Russian side is not predominantly composed of Cyrillic characters or the English side is dominated by non-Latin characters. Lastly, we combined the so obtained "in-domain" corpus with an equal amount of back-translated news data, resulting in two datasets of 2.1M sentence pairs each.

Our final submission included both deep RNN models (using multi-head and multi-hop attention with 3 heads and 2 hops) and Transformer models similar to the Transformer-Base of Vaswani et al. (2017). For the RNNs, we applied layer normalisation, label smoothing (0.1), dropout between recurrent layers (0.1), exponential smoothing and tieall embeddings. We applied similar options to our transformer models.

We trained our models in two stages: 1) Training on the full parallel corpus and 2) Fine-tuning on the "in-domain" corpus with a reduced learning rate. Each of the submitted models was optimized using the Adam algorithm, with β_1 set to 0.9, and β_2 set to 0.98. Learning rate was set to 0.0003 during the training stage and lowered to 0.00003 during the fine-tuning stage. Throughout the training, the learning rate was linearly increased over the initial 16,000 update steps up to the specified value and gradually degraded thereafter.

Model validation was performed every 5,000 steps, and we terminated training if no BLEU improvements are observed after five consecutive validations. For fine-tuning, we initialized our models with parameters corresponding to the highest validation-BLEU on the full corpus and train until convergence, as indicated by early stopping, on the "fine-tuning" training set. Due to time-constaints, convergence could not be reached for several of the ensembled models.

Our final submissions consisted of an ensemble of 4 deep RNNs for EN→RU and a mixed ensemble of 2 RNNs and 2 transformers for RU→EN. All these models were trained independently and fine-tuned on the "in-domain" set. Improvements obtained following the fine-tuning step are detailed in Table 3. While our original intention was to use mixed ensembles for both directions, our transformer models under-performed on the EN→RU translation task, which we assume is due to our hyper-parameter choices. We re-ranked the translations obtained by our left-right ensemble with a right-left ensemble of identical design. It should be noted, however, that we were unable to identify any significant improvements in terms of validation-BLEU as a result of the re-ranking. We also fine-tuned the beam-size and length penalty hyper-parameters of our ensemble systems on the corresponding validation sets for which we observe a small increase in validation-BLEU. Accordingly, we set the beam size to 20 and length normalisation parameter to 0.4 for our EN→RU ensemble and to 28 and 1.2 respectively for RU→EN.

3.7 Turkish ↔ English

After preprocessing we trained and applied a joint BPE model with 36,000 merge operations, discarding any sentences longer than 120 tokens. To produce back-translations we built systems in two steps: first we trained back-translation systems in both directions using the parallel data only, and then we re-trained them on data sets containing additional 800K back-translated sentences. Back-

	Deep RNN			Transformer		
Direction	base	fine-tuned	significance	base	fine-tuned	significance
EN→RU	30.25	32.69	$p < 0.00001$	-	-	-
RU→EN	35.79	36.5	$p < 0.005$	35.81	36.96	$p < 0.005$

Table 3: Impact of in-domain fine-tuning on the RU ↔ EN task. Reported are best validation-BLEU scores averaged over all single models of the denoted type in the submitted ensemble systems. Statistical significance was established using a paired, two-tailed t-test.

Corpus	# Synth.	R	# Total
A	800K	×1	1M
B	2.5M	×5	3.5M
C	2.5M + 1M	×5	4.5M

Table 4: Training data sets for TR↔EN systems. Data sets consist of back-translated and original parallel data oversampled R times.

translation systems are trained as deep RNN models described below. The final training sets consist of 2.5M of synthetic parallel sentence pairs created from English or Turkish NewsCrawl data sets and the SETIMES2 data oversampled 5 times (Table 4). We also experimented with copying monolingual data (Currey et al., 2017b) by adding additional 1M examples with source sentences identical to target sentences randomly selected from the monolingual data.

Our RNN models used the BiDeep architecture, and we augmented the models with layer normalisation, skip connections, and parameter tying between all embeddings and output layer. The RNN hidden state size was set to 1024, embeddings size to 512.

The architecture of transformer models was close to the Transformer-Base proposed by Vaswani et al. (2017): encoder and decoder were composed of 6 layers, and employed 8-head self-attention. We used dropout between transformer layers (0.2) as well as in attention (0.05) and feed-forward layers (0.05). The rest of parameters remained the same as in the RNN models.

Optimization used 4 GPUs with synchronous training and mini-batch size fitted into 9.5GB of GPU memory. The learning rate was linearly increased to 0.0004 reaching this value after first 18,000 updates, and then decreased by a square of the passed updates starting at 24,000 update. As a stopping criterium we used early stopping with a patience of 10 based on the word-level cross-entropy on the *newsdev2016* data sets, which served as a development set. The model was validated every 5,000 updates, and we kept best models according to the cross-entropy and BLEU score.

We evaluated systems using models with the highest BLEU score on the development set. Decoding was performed by beam search with a beam size of 12 with length normalisation with value 0.2 for EN→TR and 1.2 for TR→EN based on the greed search on the development set. Additionally, as the Turkish language is not supported by the Moses tokenizer falling back to general English tokenization rules resulting in suboptimal detokenization, we postprocessed translated Turkish texts by merging words that contains an apostrophe.

We report results on the *newstest2017* and *newstest2018* in Table 5[9]. Our first submitted TR↔EN systems were ensembles of 6 independently trained models, reranked with 3 right-left systems (Ensemble ×6 +Rerank R2L ×3). Ensembles consist of four models trained on corpus B and one model trained on corpora A and C, while each right-left model is trained on different corpora A-C. Our final systems extended the previous ensemble by 6 additional models from the same training runs that achieve best cross-entropy (instead of best BLEU) on the development set[10], utilizing 12 left-right models in total (Ensemble ×6×2). For comparison, we report the results for single systems trained on different corpora, and there is no significant performance difference among them.

3.8 Overall Performance of Submissions

In Table 6 we show the BLEU scores of our systems as compared to the top-scoring constrained systems, giving the BLEU scores from the matrix[11] and the

[9] For our official submissions we also used n-best lists generated with the beam size of 20 instead of 30, which may explain the difference between the official and reported BLEU scores.

[10] Models achieving best cross-entropy differ from the models with highest BLEU for each training run.

[11] http://matrix.statmt.org

System	EN-TR 2017	EN-TR 2018	TR-EN 2017	TR-EN 2018
Deep RNN$_A$	22.0	18.1	23.8	24.4
Deep RNN$_B$	22.1	18.6	23.9	25.1
Transformer$_A$	23.4	19.1	24.6	25.8
Transformer$_B$	23.1	19.2	25.0	26.7
Transformer$_C$	22.8	19.0	25.2	26.7
+Ensemble ×6	24.0	19.9	26.2	27.6
+Rerank R2L ×3	24.4	19.9	26.6	28.2
+Ensemble ×6×2	24.3	19.9	26.3	27.7
+Rerank R2L ×3	24.7	20.1	26.5	28.1
Submission		19.5		26.9

Table 5: Results for EN↔TR systems on official WMT test sets.

human evaluation from the findings paper (Bojar et al., 2018).

In terms of the clustering provided by the organisers, we were in the top constrained cluster (i.e. no significant difference was observed between ours and the best constrained system) for EN→CS, DE→EN, ET→EN, FI→EN, TR→EN and EN→TR, i.e. 6/14 language pairs. Nevertheless, Table 6 still shows that our systems generally lag behind the best submitted systems. This is contrast to the 2017 shared task, where we achieved the highest scores in most of the language pairs where we submitted systems. We hypothesise that other groups have taken fuller advantage of the transformer architecture, and also of data weighting and selection. We also suggest that covering all 14 language pairs meant that we had insufficient time for experimentation on some pairs, and in fact we were not able to train all models to convergence.

4 Post-Submission Experiments

In this section we present results of some post-submission experiments, which attempted to provide more insight into the contribution of different features of our system. We were especially interested in understanding why our systems tended to lag behind the performance of the best systems (in BLEU, at least). Mostly the experiments were conducted on EN↔{CS,ET,FI}.

The results are given on *newstest2017* (devtest) and *newstest2018* (test), except for ET↔EN, where devtest is half of *newsdev2018*.

4.1 Effect of Multihead/Multihop Attention

In the deep RNN models in our submissions, we used the BiDeep architecture, with multi-head/multihop attention, setting the number of hops to 3 and heads to 2. In Table 7, we show the effect of this on 3 different language pairs (both directions). For these experiments, we use the same training sets and data preparation as in our system submissions, but train the deep RNNs with a working memory of 10GB, validating every 1,000 steps, and testing for convergence with a patience of 10. We use exponential smoothing and show the results on a single smoothed model.

From the results in Table 7 we see that the multi-head/hop extension has a small positive effect on BLEU in most language pairs.

4.2 Effect of Vocabulary Size

After looking at the submission results, we questioned whether smaller vocabularies would have given better results, especially for transformer models. Having smaller vocabularies means that the models have few parameters, and also allow more words to be fitted into each training mini-batch.

To create a model with a smaller vocabulary, we follow the preparation steps used for our submissions (in EN↔{CS,ET,FI}), but use 30,000 BPE merges instead of 89,500. We show the effect both on the deep RNN model and on the Transformer model, and additionally we show the effect of tying all embeddings (i.e. source, target input and target output) on the Transformer model. The submitted models for these language pairs only have the target input and output embeddings tied. As in Section 4.1 we set the working memory for the deep RNN to 10GB, and we set the working memory for transformer training to 9.5GB. We used layer normalisation for the transformer models (although this appeared to make little if any difference to the results). In Table 8 we show the comparison for RNNs, and in Table 9 we show the same comparison for Transformer models.

Examining the results in Tables 8 and 9, we can see that the effect of vocabulary size reduction on RNN models is mixed, whereas the transformer models have a preference (in BLEU, at least) for smaller vocabularies. Tying all embeddings does not seem to help. Further investigation is needed on the vocabulary size question though, as the relationship between BPE hyper-parameters and BLEU is unclear. We note that changes in the vocabulary

	X→ EN				EN→X			
	Ours	Top	Δ BLEU	Δ DA	Ours	Top	Δ BLEU	Δ DA
CS	31.8	33.9	-3.13	-3.9	23.4	26.0	-2.59	-6.6
DE	43.9	48.4	-4.49	-4.5	44.4	48.3	-3.95	-5.6
ET	29.4	30.7	-1.30	-1.9	22.7	23.6	-0.85	-4.6
FI	23.5	24.9	-1.40	-1.2	16.7	18.2	-1.53	-5.5
RU	32.8	34.9	-2.12	-3.5	29.8	34.8	-4.95	-6.0
TR	26.9	28.0	-1.10	-1.1	19.5	20.0	-0.48	0.0
ZH	24.0	29.3	-5.31	-4.3	33.3	43.8	-10.5	-10.0

Table 6: Overall BLEU scores of our systems, compared to the top-scoring constrained systems. We also show the difference with the direct assessment (DA) score of the best constrained system.

	No hop/head		3 hop, 2 head	
Pair	devtest	test	devtest	test
CS-EN	30.0	30.8	**30.6**	**31.2**
EN-CS	23.2	23.0	**23.6**	**23.2**
ET-EN	24.8	**27.9**	**25.4**	27.2
EN-ET	**18.9**	**21.6**	18.8	21.1
FI-EN	31.4	22.6	**31.9**	**23.1**
EN-FI	24.4	16.0	**25.2**	**16.2**

Table 7: Comparison of performance of deep RNN models with/without the multihop/multihead extension.

	BPE 89.5k		BPE 30k	
Pair	devtest	test	devtest	test
CS-EN	30.6	**31.2**	**30.8**	31.1
EN-CS	**23.6**	**23.2**	23.0	22.9
ET-EN	25.4	27.2	**26.1**	**28.2**
EN-ET	**18.8**	21.1	18.7	21.1
FI-EN	**31.9**	**23.1**	31.6	22.8
EN-FI	25.2	16.2	**25.5**	**16.5**

Table 8: Effect of reducing vocabulary size for deep RNN models. We used 89,500 BPE merges for our submissions, but tried reducing it to 30,000 for post-submission experiments.

size could have a disproportionate effect on the translation of rare words (including proper nouns) which would not necessarily be detected by BLEU.

4.3 Mixed Ensembles

For our submitted systems for FI↔EN and ET↔EN we used mixed ensembles consisting of two deep RNNs and two Transformer models. In this section we examine whether the mix of architectures in the ensemble is beneficial. We compare this mixed ensemble with an ensemble of four deep RNNs.

In Table 10, we show the results. We show the mean BLEU score of the models in the ensemble, together with the overal ensemble score. For clarity, we just show scores on our test set (*newstest2018*). The gain in BLEU from ensembling (over the mean BLEU) is slightly higher in all cases than the corresponding gain for the uniform ensemble.

5 Conclusions

We have described Edinburgh's systems for all 14 language pairs, showing that we can gain improvements by augmenting a GRU-based RNN with multi-head and multi-hop attention, using mixed ensembles of deep RNNs and transformers, and selecting data from the noisy ParaCrawl corpora. Our systems perform strongly in most language pairs, except for when we did not manage to train to convergence.

Acknowledgments

This work was supported in part by funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 688139 (SUMMA), and from the Connecting Europe Facility under agreement No. NEA/CEF/ICT/A2016/1331648 (ParaCrawl).

Nikolay Bogoychev was funded by an Amazon faculty research award to Adam Lopez.

Computing resources were 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/`,

	BPE 89.5k, tied		BPE 30k, tied		BPE 30k, tied-all	
Pair	devtest	test	devtest	test	devtest	test
CS-EN	28.9	29.4	**29.4**	**29.8**	29.1	29.5
EN-CS	22.6	22.4	22.8	22.5	**23.2**	**22.6**
ET-EN	25.4	27.8	24.9	**27.9**	**25.8**	27.8
EN-ET	18.8	21.3	**19.4**	**21.9**	19.3	21.8
FI-EN	30.7	22.2	**31.3**	**22.3**	30.7	22.2
EN-FI	24.8	16.2	**25.9**	16.6	25.6	**16.7**

Table 9: Effect of reducing vocabulary size for Transformer models. We used 89,500 BPE merges for our submissions, but tried reducing it to 30,000 for post-submission experiments. We also show the effect of tying all embeddings.

	Mixed ensemble			Uniform ensemble		
Pair	mean	ensemble	Δ	mean	ensemble	Δ
ET-EN	27.7	29.0	+1.3	27.5	28.6	+1.1
EN-ET	21.5	22.7	+1.2	21.0	21.9	+0.9
FI-EN	22.5	23.2	+0.7	22.1	22.7	+0.6
EN-FI	15.9	16.5	+0.7	15.6	16.0	+0.4

Table 10: Effect of mixed versus uniform ensembles. The ensembles are either 2 deep RNNs and 2 transformers, or 4 RNNs.

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) `https://www.csd3.cam.ac.uk/getting-help/www.dirac.ac.uk%29`

- The Alan Turing Institute under the EPSRC grant EP/N510129/1.

References

Lei Jimmy Ba, Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer Normalization. *CoRR*, abs/1607.06450.

Ondřej Bojar, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, and Philipp Koehn. 2018. Findings of the 2018 conference on machine translation (WMT18). In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, Brussels, Belgium. Association for Computational Linguistics.

Mia Xu Chen, Orhan Firat, Ankur Bapna, Melvin Johnson, Wolfgang Macherey, George Foster, Llion Jones, Mike Schuster, Noam Shazeer, Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Lukasz Kaiser, Zhifeng Chen, Yonghui Wu, and Macduff Hughes. 2018. The best of both worlds: Combining recent advances in neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–86. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017a. Copied Monolingual Data Improves Low-Resource Neural Machine Translation. In *Proceedings of the Second Conference on Machine Translation, Volume 1: Research Papers*, Copenhagen, Denmark. Association for Computational Linguistics.

Anna Currey, Antonio Valerio Miceli Barone, and Kenneth Heafield. 2017b. Copied monolingual data improves low-resource neural machine translation. In *Proceedings of the Second Conference on Machine Translation*, pages 148–156, Copenhagen, Denmark. Association for Computational Linguistics.

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.

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.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, Will Lewis, Mu Li, Shujie Liu, Tie-Yan Liu, Renqian Luo, Arul Menezes, Tao Qin, Frank Seide, Xu Tan, Fei Tian, Lijun Wu, Shuangzhi Wu, Yingce Xia, Dongdong Zhang, Zhirui Zhang, and Ming Zhou. 2018. Achieving human parity on automatic chinese to english news translation.

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 ACL 2018, System Demonstrations*, pages 116–121. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *The International Conference on Learning Representations*, 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 ACL-2007 Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Marco Lui and Timothy Baldwin. 2012. langid.py: An off-the-shelf language identification tool. In *Proceedings of the ACL 2012 System Demonstrations*, pages 25–30, Jeju Island, Korea. Association for Computational Linguistics.

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

Robert C. Moore and William Lewis. 2010. Intelligent Selection of Language Model Training Data. In *Proceedings of the ACL 2010 Conference Short Papers*.

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 Second Conference on Machine Translation, Volume 2: Shared Task Papers*, Copenhagen, Denmark.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh Neural Machine Translation Systems for WMT 16. In *Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 368–373, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving Neural Machine Translation Models with Monolingual Data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.