Edinburgh Neural Machine Translation Systems for WMT 16

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

Abstract

We participated in the WMT 2016 shared news translation task by building neural translation systems for four language pairs, each trained in both directions: English↔Czech, English↔German, English↔Romanian and English↔Russian. Our systems are based on an attentional encoder-decoder, using BPE subword segmentation for open-vocabulary translation with a fixed vocabulary. We experimented with using automatic back-translations of the monolingual News corpus as additional training data, pervasive dropout, and target-bidirectional models. All reported methods give substantial improvements, and we see improvements of 4.3–11.2 BLEU over our baseline systems. In the human evaluation, our systems were the (tied) best constrained system for 7 out of 8 translation directions in which we participated.[1][2]

1 Introduction

We participated in the WMT 2016 shared news translation task by building neural translation systems for four language pairs: English↔Czech, English↔German, English↔Romanian and English↔Russian. Our systems are based on an attentional encoder-decoder (Bahdanau et al., 2015), using BPE subword segmentation for open-vocabulary translation with a fixed vocabulary (Sennrich et al., 2016b). We experimented with using automatic back-translations of the monolingual News corpus as additional training data (Sennrich et al., 2016a), pervasive dropout (Gal, 2015), and target-bidirectional models.

2 Baseline System

Our systems are attentional encoder-decoder networks (Bahdanau et al., 2015). We base our implementation on the dl4mt-tutorial[3], which we enhanced with new features such as ensemble decoding and pervasive dropout.

We use minibatches of size 80, a maximum sentence length of 50, word embeddings of size 500, and hidden layers of size 1024. We clip the gradient norm to 1.0 (Pascanu et al., 2013). We train the models with Adadelta (Zeiler, 2012), reshuffling the training corpus between epochs. We validate the model every 10 000 minibatches via BLEU on a validation set (newstest2013, newstest2014, or half of newsdev2016 for EN↔RO). We perform early stopping for single models, and use the 4 last saved models (with models saved every 30 000 minibatches) for the ensemble results. Note that ensemble scores are the result of a single training run. Due to resource limitations, we did not train ensemble components independently, which could result in more diverse models and better ensembles.

Decoding is performed with beam search with a beam size of 12. For some language pairs, we used the AmuNMT C++ decoder[4] as a more efficient alternative to the theano implementation of the dl4mt tutorial.

2.1 Byte-pair encoding (BPE)

To enable open-vocabulary translation, we segment words via byte-pair encoding (BPE)[5] (Sen-

[1] We have released the implementation that we used for the experiments as an open source toolkit: https://github.com/rsennrich/nematus

[2] We have released scripts, sample configs, synthetic training data and trained models: https://github.com/rsennrich/wmt16-scripts

[3] https://github.com/nyu-dl/dl4mt-tutorial

[4] https://github.com/emjotde/amunmt

[5] https://github.com/rsennrich/subword-nmt

Proceedings of the First Conference on Machine Translation, Volume 2: Shared Task Papers, pages 371–376,
Berlin, Germany, August 11-12, 2016.

nrich et al., 2016b). BPE, originally devised as a compression algorithm (Gage, 1994), is adapted to word segmentation as follows:

First, each word in the training vocabulary is represented as a sequence of characters, plus an end-of-word symbol. All characters are added to the symbol vocabulary. Then, the most frequent symbol pair is identified, and all its occurrences are merged, producing a new symbol that is added to the vocabulary. The previous step is repeated until a set number of merge operations have been learned.

BPE starts from a character-level segmentation, but as we increase the number of merge operations, it becomes more and more different from a pure character-level model in that frequent character sequences, and even full words, are encoded as a single symbol. This allows for a trade-off between the size of the model vocabulary and the length of training sequences. The ordered list of merge operations, learned on the training set, can be applied to any text to segment words into subword units that are in-vocabulary in respect to the training set (except for unseen characters).

To increase consistency in the segmentation of the source and target text, we combine the source and target side of the training set for learning BPE. For each language pair, we learn 89 500 merge operations.

3 Experimental Features

3.1 Synthetic Training Data

WMT provides task participants with large amounts of monolingual data, both in-domain and out-of-domain. We exploit this monolingual data for training as described in (Sennrich et al., 2016a). Specifically, we sample a subset of the available target-side monolingual corpora, translate it automatically into the source side of the respective language pair, and then use this synthetic parallel data for training. For example, for EN→RO, the back-translation is performed with a RO→EN system, and vice-versa.

Sennrich et al. (2016a) motivate the use of monolingual data with domain adaptation, reducing overfitting, and better modelling of fluency. We sample monolingual data from the News Crawl corpora[6], which is in-domain with respect to the test set.

type	DE	CS	RO	RU
parallel	4.2	52.0	0.6	2.1
synthetic ($*$ →EN)	4.2	10.0	2.0	2.0
synthetic (EN→ $*$)	3.6	8.2	2.3	2.0

Table 1: Amount of parallel and synthetic training data (number of sentences, in millions) for EN-* language pairs. For synthetic data, we separate the data according to whether the original monolingual language is English or not.

The amount of monolingual data back-translated for each translation direction ranges from 2 million to 10 million sentences. Statistics about the amount of parallel and synthetic training data are shown in Table 1. With dl4mt, we observed a translation speed of about 200 000 sentences per day (on a single Titan X GPU).

3.2 Pervasive Dropout

For English↔Romanian, we observed poor performance because of overfitting. To mitigate this, we apply dropout to all layers in the network, including recurrent ones.

Previous work dropped out different units at each time step. When applied to recurrent connections, this has the downside that it impedes the information flow over long distances, and Pham et al. (2014) propose to only apply dropout to non-recurrent connections.

Instead, we follow the approach suggested by Gal (2015), and use the same dropout mask at each time step. Our implementation differs from the recommendations by Gal (2015) in one respect: we also drop words at random, but we do so on a token level, not on a type level. In other words, if a word occurs multiple times in a sentence, we may drop out any number of its occurrences, and not just none or all.

In our English↔Romanian experiments, we drop out full words (both on the source and target side) with a probability of 0.1. For all other layers, the dropout probability is set to 0.2.

3.3 Target-bidirectional Translation

We found that during decoding, the model would occasionally assign a high probability to words based on the target context alone, ignoring the

[6]Due to recency effects, we expect last year's corpus to be most relevant, and sampled from News Crawl 2015 for EN-RO, EN-RU and EN-CS; for EN-DE, we re-used data from (Sennrich et al., 2016a), which was randomly sampled from News Crawl 2007–2014.

system	EN→DE		DE→EN	
	dev	test	dev	test
baseline	22.4	26.8	26.4	28.5
+synthetic	25.8	31.6	29.9	36.2
+ensemble	27.5	33.1	31.5	37.5
+r2l reranking	**28.1**	**34.2**	**32.1**	**38.6**

Table 2: English↔German translation results (BLEU) on dev (newstest2015) and test (newstest2016). Submitted system in bold.

source sentence. We speculate that this is an instance of the label bias problem (Lafferty et al., 2001).

To mitigate this problem, we experiment with training separate models that produce the target text from right-to-left (r2l), and re-scoring the n-best lists that are produced by the main (left-to-right) models with these r2l models. Since the right-to-left model will see a complementary target context at each time step, we expect that the averaged probabilities will be more robust. In parallel to our experiments, this idea was published by Liu et al. (2016).

We increase the size of the n-best-list to 50 for the reranking experiments.

A possible criticism of the l-r/r-l reranking approach is that the gains actually come from adding diversity to the ensemble, since we are now using two independent runs. However experiments in (Liu et al., 2016) show that a l-r/r-l reranking systems is stronger than an ensemble created from two independent l-r runs.

4 Results

4.1 English↔German

Table 2 shows results for English↔German. We observe improvements of 3.4–5.7 BLEU from training with a mix of parallel and synthetic data, compared to the baseline that is only trained on parallel data. Using an ensemble of the last 4 checkpoints gives further improvements (1.3–1.7 BLEU). Our submitted system includes reranking of the 50-best output of the left-to-right model with a right-to-left model – again an ensemble of the last 4 checkpoints – with uniform weights. This yields an improvements of 0.6–1.1 BLEU.

4.2 English↔Czech

For English→Czech, we trained our baseline model on the complete WMT16 parallel training set (including CzEng 1.6pre (Bojar et al., 2016)), until we observed convergence on our heldout set (newstest2014). This took approximately 1M minibatches, or 3 weeks. Then we continued training the model on a new parallel corpus, comprising 8.2M sentences back-translated from the Czech monolingual news2015, 5 copies of news-commentary v11, and 9M sentences sampled from Czeng 1.6pre. The model used for back-translation was a neural MT model from earlier experiments, trained on WMT15 data. The training on this synthetic mix continued for a further 400,000 minibatches.

The right-left model was trained using a similar process, but with the target side of the parallel corpus reversed prior to training. The resulting model had a slightly lower BLEU score on the dev data than the standard left-right model. We can see in Table 3 that back-translation improves performance by 2.2–2.8 BLEU, and that the final system (+r2l reranking) improves by 0.7–1.0 BLEU on the ensemble of 4, and 4.3–4.9 on the baseline.

For Czech→English the training process was similar to the above, except that we created the synthetic training data (back-translated from samples of news2015 monolingual English) in batches of 2.5M, and so were able to observe the effect of increasing the amount of synthetic data. After training a baseline model on all the WMT16 parallel set, we continued training with a parallel corpus consisting of 2 copies of the 2.5M sentences of back-translated data, 5 copies of news-commentary v11, and a matching quantity of data sampled from Czeng 1.6pre. After training this to convergence, we restarted training from the baseline model using 5M sentences of back-translated data, 5 copies of news-commentary v11, and a matching quantity of data sampled from Czeng 1.6pre. We repeated this with 7.5M sentences from news2015 monolingual, and then with 10M sentences of news2015. The back-translations were, as for English→Czech, created with an earlier NMT model trained on WMT15 data. Our final Czech→English was an ensemble of 8 systems – the last 4 save-points of the 10M synthetic data run, and the last 4 save-points of the 7.5M run. We show this as ensemble8 in Table 3, and the +synthetic results are on the last (i.e. 10M) synthetic data run.

We also show in Table 4 how increasing the amount of back-translated data affects the results.

system	EN→CS		CS→EN	
	dev	test	dev	test
baseline	18.5	20.9	23.8	25.3
+synthetic	20.7	23.7	27.2	30.1
+ensemble	22.1	24.8	28.6	31.0
+ensemble8	–	–	**29.0**	**31.4**
+r2l reranking	**22.8**	**25.8**	–	–

Table 3: English↔Czech translation results (BLEU) on dev (newstest2015) and test (newstest2016). Submitted system in bold.

system	best single		ensemble4	
	dev	test	dev	test
baseline	23.8	25.3	25.5	26.8
+2.5M synthetic	26.7	29.4	27.7	30.4
+5M synthetic	27.2	29.3	28.2	30.4
+7.5M synthetic	27.2	29.7	28.4	30.8
+10M synthetic	27.2	30.1	28.6	31.0

Table 4: Czech→English translation results (BLEU) on dev (newstest2015) and test (newstest2016), after continued training with increasing amounts of back-translated synthetic data. For each row, training was continued from the baseline model until convergence.

We see that most of the gain from back-translation comes with the first batch, but increasing the amount of back-translated data does gradually improve performance.

4.3 English↔Romanian

The results of our English↔Romanian experiments are shown in Table 5. This language pair has the smallest amount of parallel training data, and we found dropout to be very effective, yielding improvements of 4–5 BLEU.[7]

We found that the use of diacritics was inconsistent in the Romanian training (and development) data, so for Romanian→English we removed diacritics from the Romanian source side, obtaining improvements of 1.3–1.4 BLEU.

Synthetic training data gives improvements of 4.1–5.1 BLEU. for English→Romanian, we found that the best single system outperformed the ensemble of the last 4 checkpoints on dev, and we thus submitted the best single system as primary system.

[7]We also tested dropout for EN→DE with 8 million sentence pairs of training data, but found no improvement after 10 days of training. We speculate that dropout could still be helpful for datasets of this size with longer training times and/or larger networks.

system	EN→RO		RO→EN	
	dev	test	dev	test
baseline	20.2	19.2	23.6	22.7
+dropout	24.2	23.9	28.7	27.8
+remove diacritics	-	-	30.0	29.2
+synthetic	**29.3**	**28.1**	34.8	33.3
+ensemble	29.0	28.2	**35.3**	**33.9**

Table 5: English↔Romanian translation results (BLEU) on dev (newsdev2016), and test (newstest2016). Submitted system in bold.

system	EN→RU		RU→EN	
	dev	test	dev	test
baseline	21.3	20.3	22.7	22.5
+synthetic	25.8	24.3	27.1	26.9
+ensemble	**27.0**	**26.0**	**28.3**	**28.0**

Table 6: English↔Russian translation results (BLEU) on dev (newstest2015) and test (newstest2016). Submitted system in bold.

4.4 English↔Russian

For English↔Russian, we cannot effectively learn BPE on the joint vocabulary because alphabets differ. We thus follow the approach described in (Sennrich et al., 2016b), first mapping the Russian text into Latin characters via ISO-9 transliteration, then learning the BPE operations on the concatenation of the English and latinized Russian training data, then mapping the BPE operations back into Cyrillic alphabet. We apply the Latin BPE operations to the English data (training data and input), and both the Cyrillic and Latin BPE operations to the Russian data.

Translation results are shown in Table 6. As for the other language pairs, we observe strong improvements from synthetic training data (4–4.4 BLEU). Ensembles yield another 1.1–1.7 BLEU.

5 Shared Task Results

Table 7 shows the ranking of our submitted systems at the WMT16 shared news translation task. Our submissions are ranked (tied) first for 5 out of 8 translation directions in which we participated: EN↔CS, EN↔DE, and EN→RO. They are also the (tied) best constrained system for EN→RU and RO→EN, or 7 out of 8 translation directions in total.

direction	BLEU rank	human rank
EN→CS	1 of 9	1 of 20
EN→DE	1 of 11	1 of 15
EN→RO	2 of 10	1–2 of 12
EN→RU	1 of 8	2–5 of 12
CS→EN	1 of 4	1 of 12
DE→EN	1 of 6	1 of 10
RO→EN	2 of 5	2 of 7
RU→EN	3 of 6	5 of 10

Table 7: Automatic (BLEU) and human ranking of our submitted systems (uedin-nmt) at WMT16 shared news translation task. Automatic rankings are taken from `http://matrix.statmt.org` , only considering primary systems. Human rankings include anonymous online systems, and for EN↔CS, systems from the tuning task.

Our models are also used in QT21-HimL-SysComb (Peter et al., 2016), ranked 1–2 for EN→RO, and in AMU-UEDIN (Junczys-Dowmunt et al., 2016), ranked 2–3 for EN→RU, and 1–2 for RU→EN.

6 Conclusion

We describe Edinburgh's neural machine translation systems for the WMT16 shared news translation task. For all translation directions, we observe large improvements in translation quality from using synthetic parallel training data, obtained by back-translating in-domain monolingual target-side data. Pervasive dropout on all layers was used for English↔Romanian, and gave substantial improvements. For English↔German and English→Czech, we trained a right-to-left model with reversed target side, and we found reranking the system output with these reversed models helpful.

Acknowledgments

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 645452 (QT21), 644333 (TraMOOC) and 644402 (HimL).

References

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

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, Brno, Czech Republic, September 12-16, 2016, Proceedings*. Springer Verlag, September 12-16. In press.

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

Yarin Gal. 2015. A Theoretically Grounded Application of Dropout in Recurrent Neural Networks. *ArXiv e-prints*.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Rico Sennrich. 2016. The AMU-UEDIN Submission to the WMT16 News Translation Task: Attention-based NMT Models as Feature Functions in Phrase-based SMT. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Agreement on Target-bidirectional Neural Machine Translation . In *NAACL HLT 16*, San Diego, CA.

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

Jan-Thorsten Peter, Tamer Alkhouli, Hermann Ney, Matthias Huck, Fabienne Braune, Alexander Fraser, Aleš Tamchyna, Ondřej Bojar, Barry Haddow, Rico Sennrich, Frédéric Blain, Lucia Specia, Jan Niehues, Alex Waibel, Alexandre Allauzen, Lauriane Aufrant, Franck Burlot, Elena Knyazeva, Thomas Lavergne, François Yvon, and Marcis Pinnis. 2016. The QT21/HimL Combined Machine Translation System. In *Proceedings of the First Conference on Machine Translation (WMT16)*, Berlin, Germany.

Vu Pham, Théodore Bluche, Christopher Kermorvant, and Jérôme Louradour. 2014. Dropout Improves Recurrent Neural Networks for Handwriting Recognition. In *14th International Conference on Frontiers in Handwriting Recognition, ICFHR 2014, Crete, Greece, September 1-4, 2014*, pages 285–290.

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 2016), 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 2016)*, Berlin, Germany.

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