Samsung and University of Edinburgh's System for the IWSLT 2018 Low Resource MT Task

Philip Williams[2], Marcin Chochowski[1], Pawel Przybysz[1], Rico Sennrich[2], Barry Haddow[2], Alexandra Birch[2]

[1]Samsung R&D Institute, Poland
[2]School of Informatics, University of Edinburgh
{m.chochowski,p.przybysz}@samsung.com
{pwillia4,bhaddow}@inf.ed.ac.uk, {rico.sennrich,a.birch}@ed.ac.uk

Abstract

This paper describes the joint submission to the IWSLT 2018 Low Resource MT task by Samsung R&D Institute, Poland, and the University of Edinburgh. We focused on supplementing the very limited in-domain Basque-English training data with out-of-domain data, with synthetic data, and with data for other language pairs. We also experimented with a variety of model architectures and features, which included the development of extensions to the Nematus toolkit. Our submission was ultimately produced by a system combination in which we reranked translations from our strongest individual system using multiple weaker systems.

1. Introduction

This paper describes the joint submission to the IWSLT 2018 Low Resource MT task by Samsung R&D Institute, Poland, and the University of Edinburgh. We built several multilingual systems using the Tensor2Tensor[1] and Nematus[2] toolkits, ultimately choosing to use a system combination in which we reranked translations from our strongest individual system using multiple weaker systems.

As there was so little in-domain Basque-English data available, we experimented with the use of out-of-domain data, with the addition of synthetic data via back-translation, and with the incorporation of data for other language pairs. To support multilingual translation, we followed the single-model approach of [1] and simply prepended each source sentence with a token specifying the target language.

We experimented with a variety of model architectures and features. This involved the development of several extensions to the Nematus toolkit, including support for multi-GPU training, label smoothing, and mixtures of softmaxes. We have contributed our code to the public Nematus repository.

[1]https://github.com/tensorflow/tensor2tensor:1.6.3
[2]https://github.com/EdinburghNLP/nematus

2. Training Data

In brief, we used all of the provided in-domain parallel training data along with parallel data from OpenSubtitles and the Open Data Euskadil Repository. We also produced synthetic data by back-translating from English into Basque. Table 1 lists the individual parallel corpora that made up our training data. Note that not all of our systems used all of the data. We will indicate differences when describing the individual systems.

2.1. In-Domain Data

We used all of the available in-domain data, which we filtered in order to remove the TED talks covered by the devset. The task organisers had already removed the devset talks from the Basque-English training corpus, but the talks were present in the training data for all other language pairs. Since we were evaluating on the devset during system development, we filtered the in-domain data to avoid being misled by artificially strong results. In preliminary multilingual Nematus systems that used only the in-domain data, this filtering had a significant impact, reducing the BLEU score from 16.25 to 9.96.

For the Basque-Spanish and Spanish-English pairs, we used the excised training data to create supplementary devsets. Since the Basque-English devset only contained 1,140 sentence pairs, these additional devsets gave us greater confidence when evaluating system changes.

More generally, we noticed that the in-domain corpora contained many of the same talks, in effect reducing the amount of available in-domain Basque data.

2.2. Out-of-Domain Data

We added out-of-domain data for the Basque-English, Basque-Spanish, Spanish-English, and French-English language pairs. For all four, we used OpenSubtitles2018 data from the OPUS corpus. In order to avoid making our training data too unbalanced, we undersampled from the large Spanish-English and French-English corpora. This was done arbitrarily: we simply used the first N-million sentence pairs

Pair	Corpus	Sentence Pairs	eu src	es src	en tgt	es tgt
eu-en	In-domain	5,623	5,623	-	5,623	-
	OpenSubtitles2018	805,780	805,780	-	805,780	-
	Synthetic in-domain	277,097	277,097	-	277,097	-
	Synthetic OpenSubtitles2018	4,000,000	4,000,000	-	4,000,000	-
	Synthetic TED2013	1,904,674	1,904,674	-	1,904,674	-
eu-es	In-domain	5,546	5,546	-	-	5,546
	OpenSubtitles2018	793,593	793,593	-	-	793,593
	Euskadil	926,941	926,941	-	-	926,941
es-en	In-domain	277,097	-	277,097	277,097	-
	OpenSubtitles2018	10,000,000	-	10,000,000	10,000,000	-
es-fr	In-domain	277,278	-	277,278	277,278	-
eu-fr	In-domain	5,815	-	5,815	5,815	-
fr-en	In-domain	287,137	-	-	287,137	-
	OpenSubtitles2018	10,000,000	-	10,000,000	10,000,000	-
Total		29,566,581	8,719,254	20,847,327	27,840,501	1,726,080

Table 1: Statistics for the parallel training corpora used in our submission (note that not all individual systems use all of the corpora). Since we are interested in Basque-to-English translation (primarily) as well as Basque-to-Spanish and Spanish-and-English, we break down the corpus sizes for those source and target languages.

occurring in the full corpora. For Basque-Spanish, we also used the parallel data from the Open Data Euskadil Repository.

At the outset, we assumed that translation from Basque into Spanish would be easier than into English due to the greater availability of in-domain data. We contemplated pivoting from Basque to English via Spanish and therefore when selecting data from OpenSubtitles, we made an effort to include sufficient data to support high-quality Basque-Spanish and Spanish-English translation. However, translation quality for Basque-Spanish and Basque-English (as measured by BLEU) proved to be very similar and we therefore focused on direct translation from Basque to English.

As with the in-domain data, we noticed that there is a high degree of content overlap between the multilingual OpenSubtitles corpora. For OpenSubtitles2018, between 70% and 90% of the Basque side is common for Basque-English, Basque-Spanish and Basque-French making the effective size of Basque data seen by the system relatively smaller.

2.3. Synthetic Data

Basque is a language isolate spoken by less than 1 million people and as such there are few readily available parallel resources. One of the simplest ways to get more parallel data is to generate it syntheticaly through back-translation. In [2] it was shown that even poor quality synthetic corpora can improve translation quality. We used all available training data to train an English-to-Basque back-translation system for synthetic data generation (see Section 3.2 for details of the back-translation system).

In addition to back-translating the English side of the in-domain Spanish-English corpus, we back-translated the English talks from the OPUS TED2013 corpus, after filtering out the dev and test set talks. We also selected 4M pseudo in-domain sentences from OpenSubtitles using the filtering approach proposed in [3].

3. Tensor2Tensor Systems

In preliminary experiments, we tried training Transformer models [4] using both Tensor2Tensor and Marian, eventually choosing the former as the BLEU was higher. In all experiments we used the hyperparameters for *transformer_base*, setting the hidden layer size to 512, filter size to 2048, warmup steps to 16,000 and number of heads to 8. While training the back-translation model we set the *layer_prepostprocess_dropout* parameter to 0.1, while in the base systems it was set to 0.2. Each training was run on 8 GPUs for up to 300,000 training steps, with a batch size of 100 sentences per GPU.

3.1. Preprocessing

We relied on the preprocessing implemented in Tensor2Tensor for tokenization and wordpiece segmentation. For each corpus configuration we defined a new T2T problem inheriting from default TranslateProblem. We set the subword vocabulary size to 32k for all training runs, either bilingual or multilingual. The only additional preprocessing we did was punctuation normalization using the Moses toolkit and prepending the `<2xx>` tag at the beginning of source sentences, where `xx` was the code for the target language.

System	en-eu	es-eu	en-es
EnFrEs2EuFrEs	13.26	14.89	41.92

Table 2: BLEU scores for the Tensor2Tensor systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). Since this was a back-translation system, we inverted the devsets to evaluate translation in the opposite direction. This system was used for back-translation of monolingual English corpora

3.2. Back-Translation System

For back-translation, we used all of the in-domain data listed in Table 1, along with the OpenSubtitles corpora for Basque-English and Basque-Spanish, and the Euskadil corpus for Basque-Spanish. We also used 1M sentence pairs of Spanish-English, making 5.5M sentence pairs in total.

We trained both Nematus and Tensor2Tensor systems on the same dataset, obtaining results of 12.14 BLEU and 13.26 BLEU respectively on the inverted Basque-English devset (see Table 2). We chose the better-performing Tensor2Tensor system for synthetic corpus generation.

3.3. Base System

For Basque-to-English translation, we experimented with different language pair and corpus selections (Table 3). We started with a bilingual model trained only on in-domain data. Next we trained a multilingual model adding all directions for in-domain data and oversampling the Basque-English data by a factor of 20 (to better balance the larger in-domain Spanish-English corpus). This resulted in a significant improvement of almost 7 BLEU points. After adding out-of-domain and synthetic corpora we got another 5 BLEU points. Next we experimented with removing the French data from the multilingual setting as it had the least Basque sentences, giving little additional input for that language and adding complexity be adding another language into the model. We observed that removing the French parallel corpora gave significantly better results, improving Basque-English translation by 1 BLEU point on dev2018. For the final submission we used the EuEs2EnEs model for n-best list generation.

4. Nematus Systems

Nematus [5] implements a GRU-based attentional encoder-decoder. Originally based on the model in [6], the toolkit has been extended to support features such as deep architectures and input factors. Our system was based on the configuration used in University of Edinburgh's WMT17 submissions [7]. To this we added several further extensions, which we describe below.

System	eu-en	eu-es	es-en
bilingual in-domain only	11.72	-	-
EuFrEs2EnFrEs in-domain only	18.41		
+ out-of-domain	22.26	22.28	42.76
+ back-translation	23.45	17.81	43.03
EuEs2EnEs	25.09		

Table 3: BLEU scores for the Tensor2Tensor systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). The last system EuEs2EnEs was used to produce the 20-best list for further rescoring.

4.1. Preprocessing

All of our Nematus systems used a common preprocessing pipeline, consisting of five steps: normalization, tokenization, corpus cleaning, truecasing, and BPE segmentation. [8] We used scripts from the Moses toolkit [9] to perform the first four steps and subword-nmt[3] to perform the last.

The Moses tokenizer includes language-specific rules, which we opted to use.[4] However, we trained a shared truecasing model for all languages. The corpus cleaning script removes empty sentences and sentence pairs with length ratios greater than 9:1.

We trained a single joint BPE model over the full multilingual corpus, using 40,000 merge operations. Character sequences were only merged if they were observed 50 times in the training data.

4.2. Base System

Our base Nematus system used all of the data in Table 1 except for the synthetic data (which was added later for the final systems) and the French-English OpenSubtitles data. For Spanish-English we used 1M sentence pairs of OpenSubtitles rather than 10M.

4.2.1. Network Configuration

We used a word embedding size of 512 and hidden layer size of 1024. Both the encoder and decoder used a deep transition architecture [10], with 4 layers in the encoder and 8 in the decoder. We used layer normalization [11].

We tied the weights of the target-side embedding and the transpose of the output weight matrix [12]. Since the source and target sides used the same vocabulary, we also tied the source-side and target-side embeddings.

4.2.2. Training

We used the Adam [13] optimization algorithm with a learning rate of 0.0001 and a batch size of 80 (except where

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

[4] Moses does not include Basque-specific tokenization rules, so it fell back to generic tokenization for that language

noted). Training was stopped when the validation cross-entropy failed to reach a new minimum for 10 consecutive save-points (saving every 10,000 updates). The save-point used in the final model was selected based on the BLEU score of the validation set.

To speed up training, we excluded sentences in which either the source or target sentence contained more than 50 tokens.

In preliminary experiments, we found that it was important to use dropout (giving improvements of around 2 BLEU). The dropout rate was set to 0.1 for source and target word tokens and to 0.2 for embedding and hidden layers.

4.3. Extensions

4.3.1. Multi-GPU Support

Training the base model was already pushing the 12GB memory limit of our GPUs, restricting our ability to add new features. Since we did not want to risk compromising model quality by reducing the network size, we opted to implement multi-GPU training in order to reduce the per-GPU batch size, while maintaining (or increasing) the effective total batch size.[5] We added support for synchronous training in which the batch is split between multiple GPUs (on the same server), each running a full replica of the model, and then the gradients of the sub-batches are averaged. Unlike asynchronous training, this method does not affect translation quality compared to single-GPU training (assuming the batch size is constant).

4.3.2. Source language factors

As already mentioned, our training data contains tags to indicate the target language. In preliminary experiments, we found that it was beneficial to also specify the source language, which we did through the use of token-level factors. Our intuition was that the factors would help to disambiguate subword units that occur in multiple languages, but serve language-specific roles.

Since Nematus already included support for factors [14], this was simply a case of annotating the training and dev/test data with language tags and adjusting the network's word embedding settings: of the 512 source embedding units we reserved 12 for the source language factor tag and the remaining 500 for the BPE token embedding.

A contrastive experiment showing BLEU scores on dev2018 with and without source language factors can be found in Table 4.

4.3.3. Label smoothing

We implemented label smoothing [15], a regularization technique which has been shown to be effective for self-attention-based translation models [4], and, more recently, for RNN-based models similar to ours [16]. Following prior work, we set the ϵ parameter to 0.1. See Table 4 for results of a contrastive experiments with and without label smoothing.

[5] An alternative would have been to use delayed updates on a single GPU – or of course to buy GPUs with more memory.

System	eu-en	eu-es	es-en
Base	19.99	20.45	39.74
Base + source language factors	20.12	20.86	40.16
Base + label smoothing	20.46	20.65	40.16
Base + mixture of softmaxes	20.02	21.02	40.14
Base + fine-tuning	20.75	1.86	39.94

Table 4: BLEU scores for Nematus systems on dev2018 (eu-en) and on eu-es and es-en versions of the dev set (extracted from the training data). These systems use all of the parallel training data except for the synthetic data.

4.3.4. Mixture of Softmaxes

Like all standard neural translation models, our base model uses a softmax function to output a probability distribution over the target vocabulary for each timestep. For language modelling, [17] show that performance can be improved by using a combination of multiple softmax components. We reimplemented their method within Nematus and experimented with using a mixture of three softmax components. See Table 4 for results of a contrastive experiments with and without a mixture of softmaxes.

4.3.5. Fine-tuning

Since our system was trained on data drawn from multiple domains and covering several language pairs, we anticipated that there would be a benefit to fine-tuning on in-domain Basque-English data. After selecting the best model (according to validation set BLEU), we resumed training using only the in-domain Basque-English data (5,623 sentence pairs). See Table 4 for results of a contrastive experiment with and without fine-tuning.

4.4. Final Systems

Our final Nematus systems used all of the training data from Table 1, with the exception of the synthetic TED2013 corpus (since training was started before the filtered corpus was produced) and the French-English OpenSubtitles corpus. We used a 1M sentence pair version of the Spanish-English OpenSubtitles corpus. As in the Tensor2Tensor system, we oversampled the in-domain Basque-English corpus by a factor of 20. We experimented with removing the French training data but, unlike the Tensor2Tensor system, this did not improve performance (possibly because we had used less French data to start with).

We used all of the extensions just described. We trained two such systems, one using two GPUs with a total batch size of 80 and one using three GPUs with a total batch size of 160. Finally, we fine-tuned these systems giving a to-

System	dev2018	tst2018
Nematus (batch size 80)	22.56	23.18
Nematus (batch size 80, fine-tuned)	23.22	23.65
Nematus (batch size 160)	22.94	23.56
Nematus (batch size 160, fine-tuned)	23.86	24.12
Tensor2Tensor	25.09	25.40
+ reranking (default length penalty)	25.40	25.97
+ tuned length penalty	25.60	26.21

Table 5: BLEU scores on the official dev and test sets. The first five rows show the results for the individual Nematus and Tensor2Tensor systems used in the final system combination. The bottom two rows show the results of reranking the 20-best list from the Tensor2Tensor system with the Nematus systems and then tuning the length normalization parameter. The system in the bottom row is our submitted system.

tal of four Nematus systems. Unlike our base system, fine-tuning using only the in-domain data did not improve translation quality, possibly due to the oversampling of this data in the training set. Instead, we used a fine-tuning corpus that combined the genuine in-domain data with the synthetic in-domain data (which was back-translated from the English side of the Spanish-English corpus). Results with and without fine-tuning are given in Table 5.

5. System Combination

Of the individual systems, we achieved the best performance on the devset using the Tensor2Tensor EuEs2EnEs system. We used that system to generate a 20-best list, which we then rescored using the four final Nematus systems. After rescoring, we renormalized the individual scores for sentence length, optimising the length penalty (i.e., the alpha value in [18]) on dev2018, setting it to 1.5 in our submission (in all previous systems, the length penalty was set to the default value of 1.0). Finally, we reranked the list according to the sum of the five renormalized scores and used the resulting 1-best translations in our submission.

Table 5 gives BLEU scores on the dev and test sets for the five component systems and the reranked system, both with and without length penalty tuning.

6. Conclusions

For this task, we focused on supplementing the very limited in-domain Basque-to-English training data with out-of-domain data, with synthetic data, and with data for other language pairs. Through data alone, we improved translation quality from 11.72 to 25.09 BLEU.

Although our Nematus systems underperformed the Tensor2Tensor systems, we were able to narrow the gap through extensions to the base model, including label smoothing and source language factors. When evaluated on tst2018, our best Nematus system was 1.3 BLEU behind our best Tensor2Tensor system.

Despite the Nematus systems being weaker, we were able to further improve performance by reranking a 20-best list from the Tensor2Tensor system using the four final Nematus systems. Tuning the length penalty also boosted performance slightly. Our submitted system scored 26.21 BLEU on tst2018, outperforming the individual Tensor2Tensor system by 0.81 BLEU.

7. References

[1] M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, M. Hughes, and J. Dean, "Google's multilingual neural machine translation system: Enabling zero-shot translation," *Transactions of the Association for Computational Linguistics*, vol. 5, pp. 339–351, 2017.

[2] R. Sennrich, B. Haddow, and A. Birch, "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)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 86–96.

[3] A. Axelrod, X. He, and J. Gao, "Domain adaptation via pseudo in-domain data selection," in *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, ser. EMNLP '11. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011, pp. 355–362. [Online]. Available: http://dl.acm.org/citation.cfm?id=2145432.2145474

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5998–6008.

[5] R. Sennrich, O. Firat, K. Cho, A. Birch, B. Haddow, J. Hitschler, M. Junczys-Dowmunt, S. Läubli, V. A. Miceli Barone, J. Mokry, and M. Nadejde, "Nematus: a toolkit for neural machine translation," in *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 2017, pp. 65–68.

[6] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proceedings of ICLR*, 2015.

[7] R. Sennrich, A. Birch, A. Currey, U. Germann, B. Haddow, K. Heafield, A. V. Miceli Barone, and P. Williams, "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: Association for Computational Linguistics, September 2017, pp. 389–399.

[8] R. Sennrich, B. Haddow, and A. Birch, "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)*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 1715–1725.

[9] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst, "Moses: open source toolkit for statistical machine translation," in *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ser. ACL '07. Morristown, NJ, USA: Association for Computational Linguistics, 2007, pp. 177–180.

[10] A. V. Miceli Barone, J. Helcl, R. Sennrich, B. Haddow, and A. Birch, "Deep architectures for neural machine translation," in *Proceedings of the Second Conference on Machine Translation*. Copenhagen, Denmark: Association for Computational Linguistics, September 2017, pp. 99–107.

[11] L. J. Ba, R. Kiros, and G. E. Hinton, "Layer Normalization," *CoRR*, vol. abs/1607.06450, 2016.

[12] O. Press and L. Wolf, "Using the output embedding to improve language models," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. Association for Computational Linguistics, 2017, pp. 157–163.

[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *The International Conference on Learning Representations*, San Diego, California, USA, 2015.

[14] R. Sennrich and B. Haddow, "Linguistic input features improve neural machine translation," in *Proceedings of the First Conference on Machine Translation*. Berlin, Germany: Association for Computational Linguistics, August 2016, pp. 83–91.

[15] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2818–2826.

[16] M. X. Chen, O. Firat, A. Bapna, M. Johnson, W. Macherey, G. Foster, L. Jones, M. Schuster, N. Shazeer, N. Parmar, A. Vaswani, J. Uszkoreit, L. Kaiser, Z. Chen, Y. Wu, and M. Hughes, "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)*. Association for Computational Linguistics, 2018, pp. 76–86.

[17] Z. Yang, Z. Dai, R. Salakhutdinov, and W. W. Cohen, "Breaking the softmax bottleneck: A high-rank RNN language model," in *Proceedings of ICLR*, 2018.

[18] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, "Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation," *CoRR*, vol. abs/1609.08144, 2016.