FBK's Participation to the English-to-German News Translation Task of WMT 2017

Mattia A. Di Gangi[1,2] and **Nicola Bertoldi**[2] and **Marcello Federico**[2]
[1]ICT International Doctoral School - University of Trento, Italy
[2]Fondazione Bruno Kessler - Trento, Italy
{digangi,bertoldi,federico}@fbk.eu

Abstract

In this paper we report on FBK's participation to the English-to-German news translation task of the Second Conference on Machine Translation (WMT'17). The submitted system is based on Neural Machine Translation using byte-pair encoding segmentation on both source and target languages for open-vocabulary translations. Back-translations of news monolingual data are used for improving the translations fluency on the in-domain data. With respect to last year's evaluation, our baseline outperforms the 2016 best system's baseline on the test sets 2015 and 2016. However, in our set-up back-translations produced a smaller improvement than expected. The final submission is given by the combination of 7 systems, including a system trained only on true parallel data and two right-to-left systems, which improves over our single best system by 1.5 BLEU points.

1 Introduction

FBK's participation to the news translations shared task in WMT 17 focused this year on the English-German language direction. Our purpose was to explore the state of the art and build a competitive neural machine translation [3] system in order to gain a practical knowledge of the available tools. With respect to our participation in the IWSLT 2016 evaluation campaign, we switched from the Nematus-Theano framework to the OpenNMT-Torch framework [16]. The reasons were twofold: higher baseline performance and significantly faster training.

In our primary submission we used back-translations [22], BPE-encoding [23] and system combination [11]. In this paper, we report about the tools used for the submitted system and the choices we have taken in terms of hyper-parameters and used data.

The presentation is structured as follows: in Section 2 we briefly introduce the theoretical background for NMT. In Section 3 we describe our baseline system. In Sections 4 and 5 we describe the details of the back-translations and system combination, which have been used for our final submission. Evaluation results are discussed in Section 6, while Section 7 is devoted to discussion and conclusions.

2 Neural Machine Translation

Neural machine translation [25] represents the state of the art for machine translation since the outstanding results obtained on IWSLT2015 [17] IWSLT2016 [1, 7] and WMT16 [24, 5] where the neural models greatly outperformed phrase-based systems. NMT is based on the encoder-decoder-attention architecture [3] which jointly learns the translation and the alignment model with a sequence-to-sequence learning model. Given a sequence of words $f_1, f_2, \ldots, f_m$ in the source language, they are used to index an embedding look-up table and retrieve the vectors $\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_m}$ representing the words. The embeddings are processed by a bi-directional RNN

$$\overrightarrow{\mathbf{h}}_j = g(\mathbf{x}_j, \overrightarrow{\mathbf{h}}_{j-1}),\ \ j = 1, ..m$$

$$\overleftarrow{\mathbf{h}}_j = g(\mathbf{x}_j, \overleftarrow{\mathbf{h}}_{j+1}),\ \ j = m, .., 1$$

$$\mathbf{h}_j = merge(\overrightarrow{\mathbf{h}}_j, \overleftarrow{\mathbf{h}}_j)$$

where $merge$ is a function for merging the output of the RNNs, like the vector concatenation or the point-wise sum, and g is the LSTM [13] or the GRU [8] function. The sequence of vectors produced by the bidirectional RNN is the encoded

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 271–275
Copenhagen, Denmark, September 711, 2017.

representation of the source sentence.

The decoder takes as input the encoder outputs (or states) and produces a sequence of target words $e_1, e_2, \ldots, e_l$. The decoder works by progressively predicting the probability of the next target word e_i given the previously generated target words and the source context vector $\mathbf{c_i}$. At each step, the decoder computes a word embeddings $\mathbf{y}_{i-1}$ of the previous target word, applies one or more recurrent layers, an attention model function and a softmax layer. The recurrent layers produce an hidden state $\mathbf{s}_i$

$$\mathbf{s}_i = g(\mathbf{y}_{i-1}, \mathbf{s}_{i-1})$$

where, g can be computed with one or more LSTM or GRU layers. The output of the RNN is then used by the attention model to weight the source vectors according to their similarity with it.

$$\alpha_{ij} = \frac{\exp(score(\mathbf{s}_i, \mathbf{h}_j))}{\sum_{k=1}^{m} \exp(score(\mathbf{s}_i, \mathbf{h}_k))}$$

The weights are used to compute a weighted average of the encoder outputs, which represents the source context

$$\mathbf{c}_i = \sum_{j=1}^{m} \alpha_{ij} \mathbf{h}_j$$

The source context vector is then combined with the output of the last RNN layer in a new vector $\mathbf{z}_i$ that is passed as input to the softmax layer to compute the probability for each word in the vocabulary to be the next word, such that:

$$p(e \mid e_{i-1}, c_i) \propto \exp(\mathbf{e}^\top \mathbf{z}_i)$$

where $\mathbf{e}^\top$ represents the transpose of the one-hot vector representation of word e. Let Θ be the set of all the network parameters, then the objective of the training is to find parameter values maximizing the likelihood of the training set S, i.e.:

$$\sum_{(\mathbf{f},\mathbf{e}) \in S} \sum_{i=1}^{|e|} \log p(e_i | e_{<i}, \mathbf{c}_i; \Theta)$$

3 Baseline

Our baseline is a neural machine translation system trained on the four parallel corpora released for the task. Our preprocessing pipeline involved normalizing the punctuation, de-escaping the special characters, tokenization and truecasing for both English and German. We also filtered out sentence pairs with source or target length greater than 50 or length ratio in one direction more than 1:9. In Table 1 we report the number of sentences before and after the cleaning step. The last step of the preprocessing is the BPE segmentation [23]. We trained $45,000$ BPE merge rules over the joint parallel data, which resulted in a vocabulary sizes of $43,853$ words for English and $47,465$ for German.

Table 1: Number of training sentences.

	original	cleaned
commoncrawl	2,399,123	2,228,833
europarl-v7	1,920,209	1,719,859
news-comm-v12	270,769	255,944
rapid2016	1,329,041	1,277,997

The NMT architecture consists of 2 LSTM layers both in the encoder and in the decoder. We used LSTM RNNs instead of the GRU RNNs, as they performed better in our preliminary experiments. Our result is hence coherent with what reported in [6]. The word embeddings size and the number of hidden units for each LSTM layer are fixed to 500. The encoder is a bidirectional LSTM [21] with 500 hidden units equally divided among the two directions. The optimizer of choice is SGD [20] with exponential decay. In preliminary experiments, using different and smaller datasets, this optimizer outperformed Adagrad [10] and Adam [15]. Figure 1 shows the validation scores after each epoch on the validation sets with the different optimizers. In [7] Adagrad led to better results on the IWSLT En-Fr validation set, thus we argue that the choice of the optimizer depends on the dataset and the NMT implementation. The latter is not considered in studies comparing different optimizers [2]

We set the initial learning rate to 1.0 and the exponential decay to 0.9. The decay starts from epoch 9. The results of the baseline are reported in the first row of Table 3, where they are compared with our submissions. The model was trained on a single GPU for 21 epochs with a minibatch size of 120. Each epoch required about 9 hours.

4 Monolingual Data

In order to leverage monolingual data we followed the state-of-the-art practice of using back-

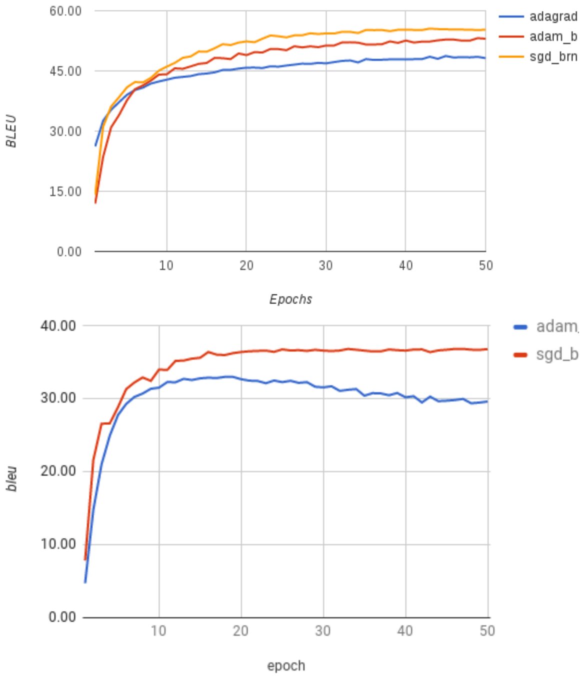

Figure 1: Comparison between different optimizers in terms of BLEU. In the top Figure SGD with exponential decay is the best performing against Adam and Adagrad in a private dataset. In the bottom Figure the trend is confirmed on IWSLT EN-FR data.

translated data. A German-to-English MT system was used to translate the news monolingual sentences. As we did not plan to participate in the opposite direction, we decided to use a phrase-based MT to performing back-translations.

The system of choice was MMT [4], an open-source PBMT system for industrial use, which has been trained using all available parallel data. The language model was trained on sentences randomly sampled from the English monolingual newscrawl data for a total of 1B words. The log-linear model weights were tuned on 1000 sentences sampled from newstest2013 and newstest2014. After tuning, the system obtained a BLEU score of 25.04 on newstest2015 de-en. With ModernMT we were able to translate $250,000$ sentences per day on a single CPU. We translated in total about $30M$ newscrawl sentences from 2013 to 2016. In a first experiment we trained a model until convergence on this huge synthetic parallel data and then fine-tuned on the true parallel data. In this setting, the system trained on the synthetic data converged before finishing the first epoch, and the following fine-tuning reached only 23 Bleu scores on newstest2015, thus we decided not to use this data for the final submission. Our best single system continued the training of the baseline on a new dataset consisting of both the parallel sentences and $5M$ back-translated parallel sentences randomly sampled from the $30M$ set.

Table 2: Results of the single systems used for combination

System	newstest2015	newstest2016
Sys1	23.95	28.53
Sys2	25.41	29.68
Sys3	**25.69**	**30.21**
Sys4	25.26	28.69

As we describe in the following section, we used monolingual data also for the system combination.

5 System Combination

Our primary submission has been produced by merging the outputs of different systems with Jane's system combination tool [11].

For a system combination of m systems we build m confusion networks that are then merged to form a single confusion network. For each of the small networks, only one of the systems is chosen as the primary system, which is the system that decides the word order. The sentences from every secondary systems are then aligned to the primary. We perform word alignment using METEOR [9], a tool that uses four criteria for aligning words: 1) exact match; 2) stem, which matches two words if their stems computed with the Snowball Stemmer [19] are the same; 3) synonym, which uses the WordNet [18] synsets database; 4) paraphrase, which matches phrases if they are in an internal paraphrase table. When no criterion is matched, there is a match with the empty string.

The confusion networks are initialized with the primary system sentences, then the words from the secondary hypothesis are added to the network according to the alignment. The final confusion network is obtained by the union of the m networks. The output sentence is produced from the confusion network by majority voting. Each hypothesis receives a system weight, and the weights are optimized using a development set. In our case the development set is newstest2015 and the validation set is newstest2016

The systems involved in the combination are from

System	2015	2016	2017
Baseline (sys4)	25.26	28.69	24.20
+ Synthetic (sys3)	25.69	30.21	24.80
System Combination	**28.10***	**32.84**	**26.30**

Table 3: Our results on newstest 2015-17.
*The system has been tuned on newstest2015.

4 different NMT systems that used different training data:

1. A NMT system trained on parallel + synthetic[1] for 12 epochs
2. An NMT trained on parallel + synthetic right to left for 11 epochs[2]
3. The tuning of the baseline for 7 epochs more on parallel + synthetic data
4. The baseline system

For each system, with the exception of the baseline, we used the weights of last two epochs. This gave us an improvement on the validation set of 0.5 Bleu points. We improved the system combination by adding a 5-grams language model with modified Kneser-Ney smoothing [14] without pruning, trained on $\sim$ 500M tokens with KenLM [12]. This improved the result by another +0.6 BLEU on the validation.

In Table 2 we present the results of the single systems on newstest 2015 and 16. As expected, the systems are quite different also in terms of performance, especially for newstest2016, thus we expected significant improvements.

Surprisingly, we found that our system trained from scratch on back-translated data performed worse than the baseline, while the right-to-left system trained on the same data is slightly better on newstest2015 and 1 Bleu point better on newstest2016. The best system is the one that was trained in two phases, during the first phase only on true parallel data, and continued after 21 epochs on true plus synthetic parallel sentences.

6 Results

In Table 3 we report the results in terms of Bleu scores, for the test sets from 2015 to 2017. On newstest2015 the baseline was already in par with last year's best single system [24], and the improvement obtained by back-translations is only of +0.4 Bleu scores. The improvement given by back-translations is more significant on newstest2016, for which our system was quite weak if compared with last year's best single system, and it improved by +1.6 Bleu. The improvement is small also for newstest2017, where it amounts to $+0.6$.

In the last row of the table the results of the system combination are reported. For newstest2015 we get an improvement of $+2.4$, but the weights are optimized according to this dataset. A similar improvement is obtained on newstest2016, where we gain $+2.6$ Bleu scores. The improvement is considerable but the best single system does not have state-of-the-art results on this dataset. On newstest2017 the improvement over our best single system is of $+1.5$ Bleu scores, thus it produced a final score of 26.30 for which it has been ranked 8th out of 21 systems.

From Tables 2 and 3 we can see that the back-translations gained a small improvement to our systems, specially when there has not been a previous training over only true parallel data (sys1 in Table 2). This is surely related to the number of back-translated sentences, which was maybe too high with respect to the number of parallel sentences. Another issue can be due to the quality of the back-translations that were done with a PBMT system, hence underperforming with respect to a state-of-the-art NMT system.

7 Conclusions

In this paper we have reported on our submission to the English-German news translation task of WMT17. We developed several NMT systems with the OpenNMT open-source tool that were trained over real and synthetic parallel data. We used BPE segmentation for open-vocabulary translation and back-translations to create additional synthetic translations. The best single system, trained on true parallel data and afterwards on true and synthetic parallel sentence pairs, obtained state-of-the-art results on newstest2015 but not on newstest2016 and newstest2017. Additional data created via back-translations did not pay off as hoped. The outputs of 4 different systems, including a right-to-left system, were combined using system combination, producing an improvement

[1] With synthetic we refer to 5M back-translated sentences randomly sampled from newscrawl.

[2] In a right-to-left system the target sentences are in reverse order.

of +1.5 BLEU on this year's test set.

Acknowledgments

This work has been partially supported by the EC-funded projects ModernMT (H2020 grant agreement no. 645487) and QT21 (H2020 grant agreement no. 645452).

References

[1] The University of Edinburgh's systems submission to the MT task at IWSLT, author=Junczys-Dowmunt, Marcin and Birch, Alexandra, booktitle=Proceedings of the ninth International Workshop on Spoken Language Translation (IWSLT), Seattle, WA, year=2016.

[2] P. Bahar, T. Alkhouli, J.-T. Peter, C. J.-S. Brix, and H. Ney. Empirical investigation of optimization algorithms in neural machine translation. *The Prague Bulletin of Mathematical Linguistics*, 108(1):13–25, 2017.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[4] N. Bertoldi, R. Cattoni, M. Cettolo, M. A. Farajian, M. Federico, D. Caroselli, L. Mastrostefano, A. Rossi, M. Trombetti, U. Germann, and D. Madl. MMT: New open source MT for the translation industry. In *Proceedings of The 20th Annual Conference of the European Association for Machine Translation (EAMT)*, 2017.

[5] J. Bradbury and R. Socher. Metamind neural machine translation system for WMT 2016. In *Proceedings of the First Conference on Machine Translation, Berlin, Germany. Association for Computational Linguistics*, 2016.

[6] D. Britz, A. Goldie, T. Luong, and Q. Le. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*, 2017.

[7] R. Chatterjee, M. Farajian, C. Conforti, S. Jalalvand, V. Balaraman, M. Di Gangi, D. Ataman, M. Turchi, M. Negri, and M. Federico. FBK's neural machine translation systems for IWSLT 2016. In *Proceedings of 13th International Workshop on Spoken Language Translation (IWSLT 2016)*, 2016.

[8] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*, 2014.

[9] M. Denkowski and A. Lavie. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91. Association for Computational Linguistics, 2011.

[10] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[11] M. Freitag, M. Huck, and H. Ney. Jane: Open source machine translation system combination. In *EACL*, pages 29–32, 2014.

[12] K. Heafield, I. Pouzyrevsky, J. H. Clark, and P. Koehn. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August 2013.

[13] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[14] F. James. Modified kneser-ney smoothing of n-gram models. *Research Institute for Advanced Computer Science, Tech. Rep. 00.07*, 2000.

[15] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[16] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush. OpenNMT: Open-Source Toolkit for Neural Machine Translation. *ArXiv e-prints*.

[17] M.-T. Luong and C. D. Manning. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, 2015.

[18] G. A. Miller and C. Fellbaum. Wordnet then and now. *Language Resources and Evaluation*, 41(2):209–214, 2007.

[19] M. F. Porter. Snowball: A language for stemming algorithms, 2001.

[20] H. Robbins and S. Monro. A stochastic approximation method. *The annals of mathematical statistics*, pages 400–407, 1951.

[21] M. Schuster and K. K. Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[22] R. Sennrich, B. Haddow, and A. Birch. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*, 2015.

[23] R. Sennrich, B. Haddow, and A. Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

[24] R. Sennrich, B. Haddow, and A. Birch. Edinburgh neural machine translation systems for WMT 16. *arXiv preprint arXiv:1606.02891*, 2016.

[25] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.