Multi-source Transformer with Combined Losses for Automatic Post-Editing

Amirhossein Tebbifakhr[1,2], **Ruchit Agrawal**[1,2], **Rajen Chatterjee**[1,2]
Matteo Negri[1], **Marco Turchi**[1]
[1] Fondazione Bruno Kessler, Via Sommarive 18, Povo, Trento - Italy
[2] University of Trento, Italy
{atebbifakhr,ragrawal,chatterjee,negri,turchi}@fbk.eu

Abstract

Recent approaches to the Automatic Post-editing (APE) of Machine Translation (MT) have shown that best results are obtained by neural multi-source models that correct the raw MT output by also considering information from the corresponding source sentence. To this aim, we present for the first time a neural multi-source APE model based on the Transformer architecture. Moreover, we employ sequence-level loss functions in order to avoid exposure bias during training and to be consistent with the automatic evaluation metrics used for the task. These are the main features of our submissions to the WMT 2018 APE shared task (Chatterjee et al., 2018), where we participated both in the PBSMT subtask (i.e. the correction of MT outputs from a phrase-based system) and in the NMT subtask (i.e. the correction of neural outputs). In the first subtask, our system improves over the baseline up to -5.3 TER and +8.23 BLEU points ranking second out of 11 submitted runs. In the second one, characterized by the higher quality of the initial translations, we report lower but statistically significant gains (up to -0.38 TER and +0.8 BLEU), ranking first out of 10 submissions.

1 Introduction

The purpose of Automatic Post-Editing (APE) is to correct the raw output of a Machine Translation system by learning from human corrections. Since the inner workings of MT engines are often not accessible (e.g. by users relying on Google Translate), hence impossible to modify and improve, APE becomes a solution to enhance the quality of the translated segments. Good solutions to the problem have high potential in the translation industry, where better translation means lower costs for human revision and where the adaptations of third-party, general-purpose systems to new projects is a major need.

In the last few years, the APE shared tasks at WMT (Bojar et al., 2015, 2016, 2017) have renewed the interests in this topic and boosted the technology around it. Moving from the phrase-based approaches used in the first editions of the task (Chatterjee et al., 2015), last year the multi-source neural models (Chatterjee et al., 2017; Junczys-Dowmunt and Grundkiewicz, 2017; Hokamp, 2017) have shown their capability to significantly improve the output of a PBSMT system. These APE systems shared several features and implementation choices, namely: *1)* an RNN-based architecture, *2)* the use of large artificial corpora for training, *3)* model ensembling techniques, *4)* parameter optimization based on Maximum Likelihood Estimation (MLE) and *5)* vocabulary reduction using the Byte Pair Encoding (BPE) technique. Although they achieve good performance and impressive translation quality improvements, some of these techniques are not optimal for the actual deployment of APE technology in the translation industry. The main reasons are the long time required for model training and the high maintenance costs of complex architectures that combine multiple models. To make APE solutions usable and useful for the industrial market, our submissions focus on the development of an end-to-end system that does not require multiple models and external components (e.g. hypothesis re-ranker), but leverages a fast to train architecture, effective pre-processing methods and task-specific losses to boost performance. Our main contributions are:

- We adapt the Transformer (Vaswani et al., 2017) to the APE problem, so that multiple encoders can be exploited to leverage information both from the MT output to be corrected and from the corresponding source sentence (multi-source encoding).

Proceedings of the Third Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 859–865
Brussels, Belgium, October 31 - Novermber 1, 2018. ©2018 Association for Computational Linguistics
https://doi.org/10.18653/v1/W18-64099

- We explore different strategies for combining token and sentence level losses.
- We apply *ad hoc* pre-processing for the German language by re-implementing the pipeline used by the best system at the WMT'17 Translation task (Huck et al., 2017).
- In addition to the artificial data released by (Junczys-Dowmunt and Grundkiewicz, 2016), we make extensive use of a synthetic corpus of 7.2M English-German triplets (Negri et al., 2018), which was provided by the organizers as additional training material.

We participated in both the APE'18 subtasks with positive results. In the PBSMT subtask our top run improves the baseline up to -5.3 TER and +8.23 BLEU points (ranking second out of 11 submissions) while, in the NMT subtask, it achieves a -0.38 TER and +0.8 BLEU improvement (ranking first out of 10 submissions).

2 Multi-source Transformer Network

The Transformer network (Vaswani et al., 2017), like most of the sequence-to-sequence models, follows an encoder-decoder architecture. It uses stacked layers for the encoder and the decoder. The encoder layers consist of a multi-head self-attention, followed by a position-wise feed-forward network. The decoder layers have an extra multi-head encoder-decoder attention after the multi-head self-attention sub-layer. Also, a softmax normalization is applied to the output of the last layer in the decoder to generate a probability distribution over the target vocabulary. Since there is no recurrence in this architecture, a positional encoding is added to both the source and the target word embeddings in order to empower the model to capture the position of the words. More formally, the positional encoding is defined as follows:

$$\text{PE}(pos, 2i) = \sin(pos/10000^{2i/d_{model}})$$
$$\text{PE}(pos, 2i+1) = \cos(pos/10000^{2i/d_{model}})$$

where pos is the position of the word in the sentence, i is the dimension of the vector, and d_{model} is the dimensionality of the word embeddings.

The attention is a mapping from a query (Q), a key (K), and a value (V) to an output vector. In Transformer, the attention is based on dot-product attention which is defined as follow:

$$\text{Attention}(Q, K, V) = \text{softmax}(QK^T/\sqrt{d_k})V$$

where d_k is added as a scaling factor for improving the numerical stability, which is equal to the dimensionality of the key matrix. The multi-head attention receives h different representations of (Q, K, V), which makes it possible to learn different relationships between information coming from different positions simultaneously. It is computed as follows:

$$\text{MH}(Q, K, V) = \text{Concat}(head_1, ..., head_h)W^O$$

where h is number of heads and W^O is a parameter matrix with $hd_v{*}d_{model}$ dimension. In Transformer, the multi-head attention is used in two different ways: encoder-decoder and self-attention. In the self-attention, in both the encoder and the decoder, the Q, K, and V matrices are coming from the previous layer, while in the encoder-decoder attention, Q matrix comes from the previous layer, and the K and V matrices come from the encoder.

In order to encode the source sentence in addition to the MT output, we employ the multi-source method by Zoph and Knight (2016). Our model consists of two encoders, one for the source sentence and one for the MT output. The outputs of these two encoders are concatenated and passed as the key in the attention. This helps to have a better representation, leading to a more effective attention at decoding time.

3 Sequence-Level Loss Function

For training the model, most of the approaches in sequence-to-sequence modeling try to maximize the likelihood over the training data. In this scenario, the loss function is a token-level loss defined as:

$$\mathcal{L}_{\text{MLE}} = -\sum_{n=1}^{N} p(y_n|y_{<n}, \mathbf{x})$$

where $p(y_n|y_{<n}, \mathbf{x})$ is the probability of generating the target word in the n-th position. Ranzato et al. (2015), however, indicate two drawbacks for Maximum Likelihood Estimation (MLE). First, during training, the previous words passed to the decoder are always chosen from the ground-truth. However, the fact that at test time the previous

words are chosen from the model distribution, results in a bias called *exposure bias*. Such bias makes the model unable to recover from the errors made in the decoding step, which easily have a cumulative catastrophic effect. Second, using MLE as loss function, the model is optimized to maximize the probability of the training data, while the performance of the model is evaluated by the sequence-level evaluation metrics (TER and BLEU in the case of the APE task). In order to overcome the mentioned drawbacks, following Minimum Risk Training (MRT) introduced by Shen et al. (2016), we use a risk function which is defined as:

$$\mathcal{R}_{\text{MRT}} = \sum_{\mathbf{y}\in\mathcal{S}(\mathbf{x})} \frac{P(\mathbf{y}|\mathbf{x})}{\sum_{\mathbf{y}'\in\mathcal{S}(\mathbf{x})} P(\mathbf{y}'|\mathbf{x})} \Delta(\mathbf{y})$$

where $\mathcal{S}(\mathbf{x})$ is a set of sampled hypotheses from the model for the input sentence $\mathbf{x}$, $P(\mathbf{y}|\mathbf{x})$ is the probability of the sampled hypothesis, and $\Delta(\mathbf{y})$ is a cost value for generating the sample $\mathbf{y}$, e.g. $\Delta(\mathbf{y}) = -\text{BLEU}(\mathbf{y})$. Following Sennrich et al. (2016), we employ negative smoothed sentence-level BLEU (Papineni et al., 2002; Chen and Cherry, 2014) for computing the cost function.[1]

4 Data Pre-processing

In order to reduce the vocabulary size, on the German MT output and post-edits we apply our reimplementation of the word segmentation method introduced by Huck et al. (2017). It consists of three different steps:

1. Suffixes are separated from the word stems by using a modified version of snowball stemming, which separates and keeps the suffixes instead of stripping them;

2. The output of the previous step is passed to the empirical compound splitter described in (Koehn and Knight, 2003), which is run with the same parameters reported in (Huck et al., 2017);

3. The output of the previous step is segmented with Byte Pair Encoding (BPE) (Sennrich et al., 2016).

[1]Although TER (Snover et al., 2006) is the primary evaluation metric for the task, we opted for BLEU since, according to (Shen et al., 2016), optimizing with this metric gives better results also when evaluation is done with TER.

For the English source sentences, we only use BPE to reduce the vocabulary size.

5 Experimental Setting

5.1 Data

To train our models, we used both the in-domain data released by APE the task organizers and the synthetic data provided as additional training material.

In-domain Data. In-domain data consist of English-German (SRC, MT, PE) triplets in which the MT element (a German translation of the English SRC sentence) has been generated by "black-box" MT systems: a phrase-based one for the PBSMT subtask and a neural one for the NMT subtask. In both cases, the post-edit element (PE) is a correction of the target made by professional post-editors. The PBSMT training set, which is the largest one, comprises 28K triplets. The NMT training set, is smaller in size and contains 13K instances. From the two training corpora, we extracted 1K triplets to be used as development set to compare the performance of different models during training.

Synthetic data. Since building neural APE models heavily relies on the availability of large training data, we took advantage of the following two corpora:

- the eSCAPE corpus (Negri et al., 2018), which contains 7.2M English-German triplets for each MT paradigm (i.e. 7.2M phrase-based and neural translations of the same source sentences). It has been generated from a parallel English-German corpus, by taking the target sentences as artificial post-edits and the machine-translated source sentences as MT elements of each triplet.

- The artificial corpus provided by Junczys-Dowmunt and Grundkiewicz (2016), which contains 4.0M English-German triplets generated by applying a round-trip translation protocol to German monolingual data.

Before applying the pre-processing described in Section 4 to the eSCAPE data, we performed the following two cleaning steps:

1. We removed the triplets in which the length ratio between the source sentence and the

post-edit is too different from the average in the corpus;

2. We run a language identifier[2] in order to remove the triplets having a non-English source sentence or a non-German post-edit.

The application of these two cleaning steps resulted in the removal of approximately 600K instances from the eSCAPE corpus.

5.2 Evaluation Metrics

In order to evaluate our models, we use the two automatic evaluation metrics: i) TER which is computed based on edit distance (Snover et al., 2006) and ii) BLEU which is the geometric mean of n-gram precisions multiplied to the brevity penalty (Papineni et al., 2002).

5.3 Hyperparameters

We set the number of merging rules to 32K for applying BPE in the pre-processing steps. We employ OpenNMT-tf toolkit (Klein et al., 2017) for our implementation, by using 512 dimensions for word embeddings, 4 layers for both the encoders and the decoder with 512 units, and feed-froward dimension of 1,024. In order to avoid over-fitting, we use attention and residual dropout by setting the dropout probability to 0.1, along with the label-smoothing with parameter equal to 0.1. For training using MLE, we use the Adam optimizer (Kingma and Ba, 2014) with batch size of 8,192 tokens, learning rate of 2.0 and the warm-up strategy introduced by (Vaswani et al., 2017) with the warm-up steps equals to 8,000. For training using MRT, we use stochastic gradient descent optimizer with the batch size of 4,096 tokens. We also employ the beam search with beam width of 5 to sample hypotheses from the model.

6 Results

For both the subtasks, we train six different models. The performance of these models on the PBSMT and NMT development sets is reported in Tables 1 and 2.

Generic. First, we train a model using the union of the (out-domain) synthetic datasets. As expected, the performance of this model in both subtasks is lower than the baseline. We only train this model as initial generic model in order to fine-tune it using the in-domain data.

MLE. Using MLE, we fine-tune the generic model on the corresponding in-domain data for each subtask. For the PBSMT subtask, this model achieves a -6.73 TER and +9.94 BLEU improvement over the baseline. The gain is much lower for the NMT subtask (-0.33 TER and +0.85 BLEU), confirming that, together with the availability of less training data, the quality of the underlying NMT system has left little space for improvement.

MRT. We continue the training by using MRT in two ways: *i)* by adding the reference to the set of hypotheses sampled from the model and *ii)* without adding the reference. In contrast with Shen et al. (2016), who suggest to add the reference to the sampled set of hypotheses, we found that adding the reference is harmful. Actually, by adding the reference to the sample, the other hypotheses are considered as poor alternatives, since they have a lower BLEU score. Nevertheless, these samples usually have good quality and a considerable overlap with the reference. Therefore, updating the model in the direction of decreasing the probability of these hypotheses is does not seem a promising direction.

MRT + MLE. In order to avoid this problem and take advantage of the reference, we re-run the previous learning step using the linear combination of the two loss functions. Formally, we use the following loss function:

$$\mathcal{L}_{comb} = \alpha\mathcal{L}_{\text{MLE}} + (1-\alpha)\mathcal{R}_{\text{MRT}}$$

where α is set to 0.5 to give equal importance to the two components[3] The results show that combining the two loss functions makes the model able to learn also from the reference without ignoring the contribution of the other hypotheses.

Our model outperforms the best performing system at the last round of the shared task (Chatterjee et al., 2017), with improvements up to -1.27 TER and +1.23 BLEU on the PBSMT development set. Although we are using more out-of-domain data, it is interesting to note that these scores are obtained with a much simpler architecture, which does not require to ensemble n models and to train a re-ranker. Since using only

[2]For this purpose we used the language detector available at: https://github.com/optimaize/language-detector.

[3]We leave for future work the empirical estimation of optimal values for α.

Model	Reference	TER	BLEU
Generic	-	25.00	61.69
MLE	-	18.08	72.86
MRT	Yes	18.02	72.91
MRT	No	18.44	73.05
MLE + MRT	Yes	17.99	72.99
MLE + MRT	No	**17.95**	**73.12**^{1}

Table 1: Results of the multi-source Transformer with specific losses on the PBSMT outputs. The performance of the MT baseline are 24.81 TER and 62.92 BLEU. Superscript 1 denotes that improvement over MLE is statistically significant.

Model	Reference	TER	BLEU
Generic	-	17.35	72.55
MLE	-	14.75	77.61
MRT	Yes	14.81	77.57
MRT	No	14.78	**77.74**
MRT + MLE	Yes	14.75	77.68
MRT + MLE	No	**14.68**	77.68

Table 2: Results of the multi-source Transformer with specific losses on the NMT outputs. The performance of the MT baseline are 15.08 TER and 76.76 BLEU.

MRT produced a better BLEU score in NMT subtask, we submitted the best model using only MRT without reference as our *Primary* submission, and the best model using MRT+MLE as our *Contrastive* submission.

7 Test Set Results

The performance of the primary and contrastive APE systems on the test set for both the subtasks is reported in Table 3. Apart from a minimal variation in the TER scores for the PBSMT subtask, the results confirm what previously seen on the development set. Our APE systems are able to significantly improve the quality of the PBSMT outputs by achieving a gain of -5.62 TER and +8.23 BLEU points.

When post-editing the output of a NMT system, the gains are smaller (-0.38 TER and +0.8 BLEU). This is somehow expected since the role of an APE system is to fix MT errors: in presence of higher quality translations (from PBSMT to NMT: -7.4 TER and +12.54 BLEU) there are less errors to correct and the chance to apply unnecessary changes is higher. Apart from that, our APE systems are able to improve the NMT outputs showing that, even in this challenging condition,

Task	System	TER	BLEU
PBSMT	Baseline	24.24	62.99
	Primary	18.94	**71.22**
	Contrastive	**18.62**	71.04
NMT	Baseline	16.84	74.73
	Primary	**16.46**	**75.53**
	Contrastive	16.55	75.38

Table 3: Submissions at the WMT APE shared task.

APE is useful.

8 Conclusion

We presented the FBK's submissions to the APE shared task at WMT 2018 (Chatterjee et al., 2018). Our models extend a Transformer-based architecture by: *1)* leveraging multi-source inputs consisting in the source and MT texts and *2)* taking advantage of combined token and task-specific losses. Moreover, an *ad hoc* text pre-processing for the German language and more artificial data are exploited to help the training of the model. The resulting systems show large gains in performance when post-editing the PBSMT translations (our top-submission ranks second in this subtask), while minimal improvements are obtained when correcting the NMT outputs (still, our top-run ranks first in this subtask). These differences in performance strongly depend on the initial quality of the MT outputs that significantly changes from the PBSMT to the NMT system.

It is worth to remark that our implementation choices were mainly driven by the needs of a translation market in which simple solutions that are easy to maintain are always preferable to complex architectures. In this direction, our APE systems consist of a single network that can be trained in an end-to-end fashion, without recourse to ensembles of multiple models or the concatenation of components (e.g. hypothesis re-ranker) that have to be trained independently.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Shujian Huang, Matthias Huck, Philipp Koehn, Qun Liu, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Raphael Rubino, Lucia Specia, and Marco Turchi. 2017. Findings of the 2017 Conference on Machine Translation (WMT17). In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 169–214, Copenhagen, Denmark. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurelie Neveol, Mariana Neves, Martin Popel, Matt Post, Raphael Rubino, Carolina Scarton, Lucia Specia, Marco Turchi, Karin Verspoor, and Marcos Zampieri. 2016. Findings of the 2016 Conference on Machine Translation. In *Proceedings of the First Conference on Machine Translation*, pages 131–198, Berlin, Germany. Association for Computational Linguistics.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Barry Haddow, Matthias Huck, Chris Hokamp, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Matt Post, Carolina Scarton, Lucia Specia, and Marco Turchi. 2015. Findings of the 2015 Workshop on Statistical Machine Translation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 1–46, Lisbon, Portugal. Association for Computational Linguistics.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source Neural Automatic Post-Editing: FBK's participation in the WMT 2017 APE shared task. In *Proceedings of the Second Conference on Machine Translation*, pages 630–638. Association for Computational Linguistics.

Rajen Chatterjee, Matteo Negri, Raphael Rubino, and Marco Turchi. 2018. Findings of the WMT 2018 Shared Task on Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Belgium, Brussels. Association for Computational Linguistics.

Rajen Chatterjee, Marco Turchi, and Matteo Negri. 2015. The FBK Participation in the WMT15 Automatic Post-editing Shared Task. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 210–215, Lisbon, Portugal. Association for Computational Linguistics.

Boxing Chen and Colin Cherry. 2014. A systematic comparison of Smoothing Techniques for Sentence-level BLEU. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 362–367.

Chris Hokamp. 2017. Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 647–654, Copenhagen, Denmark. Association for Computational Linguistics.

Matthias Huck, Fabienne Braune, and Alexander Fraser. 2017. LMU Munich's Neural Machine Translation Systems for News Articles and Health Information Texts. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 315–322, Copenhagen, Denmark. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2016. Log-linear Combinations of Monolingual and Bilingual Neural Machine Translation Models for Automatic Post-Editing. In *Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers*, pages 751–758. Association for Computational Linguistics.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2017. The AMU-UEdin Submission to the WMT 2017 Shared Task on Automatic Post-Editing. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 639–646, Copenhagen, Denmark. Association for Computational Linguistics.

D. P. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *ArXiv e-prints*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-Source Toolkit for Neural Machine Translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada.

Philipp Koehn and Kevin Knight. 2003. Empirical Methods for Compound Splitting. In *Proceedings of the Tenth Conference on European Chapter of the Association for Computational Linguistics - Volume 1*, EACL '03, pages 187–193, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matteo Negri, Marco Turchi, Rajen Chatterjee, and Nicola Bertoldi. 2018. eSCAPE: a Large-scale Synthetic Corpus for Automatic Post-Editing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Philadelphia, Pennsylvania. Association for Computational Linguistics.

M. Ranzato, S. Chopra, M. Auli, and W. Zaremba. 2015. Sequence Level Training with Recurrent Neural Networks. *ArXiv e-prints*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas*, pages 223–231, Cambridge, Massachusetts, USA.

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.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. *arXiv preprint arXiv:1601.00710*.