Online Neural Automatic Post-editing for Neural Machine Translation

Matteo Negri[1] **Marco Turchi**[1] **Nicola Bertoldi**[1,2] **Marcello Federico**[1,2]

[1] Fondazione Bruno Kessler - Trento, Italia
[2] MMT Srl - Trento, Italia
[negri,turchi,bertoldi,federico]@fbk.eu

Abstract

English. Machine learning from user corrections is key to the industrial deployment of machine translation (MT). We introduce the first on-line approach to automatic post-editing (APE), i.e. the task of automatically correcting MT errors. We present experimental results of APE on English-Italian MT by simulating human post-edits with human reference translations, and by applying online APE on MT outputs of increasing quality. By evaluating APE on generic vs. specialised and static vs. adaptive neural MT, we address the question: At what cost on the MT side will APE become useless?

Italiano. *L'apprendimento automatico dalle correzioni degli utenti è fondamentale per lo sviluppo industriale della traduzione automatica (MT). In questo lavoro, introduciamo il primo approccio on-line al post-editing automatico (APE), ovvero il compito di correggere automaticamente gli errori della MT. Presentiamo risultati di online APE su MT da inglese a italiano simulando le correzioni umane con traduzioni manuali già disponibili e utilizzando MT di qualità crescente. Valutando l'APE su MT neurale generica oppure specializzata, statica o adattiva, affrontiamo la domanda di fondo: a fronte di quale costo sul lato MT l'APE diventerà inutile?*

1 Introduction

Automatic Post-editing for MT is a supervised learning task aimed to correct errors in a machine-translated text (Knight and Chander, 1994; Simard et al., 2007). Cast as a problem of "monolingual translation" (from raw MT output into improved text in the same target language), APE has followed a similar evolution to that of MT. As in MT, APE research received a strong boost from shared evaluation exercises like those organized within the well-established WMT Conference on Machine Translation (Chatterjee et al., 2018). In terms of approaches, early MT-like phrase-based solutions (Béchara et al., 2011; Rosa et al., 2013; Lagarda et al., 2015; Chatterjee et al., 2015) have been recently outperformed and replaced by neural architectures that now represent the state of the art (Junczys-Dowmunt and Grundkiewicz, 2016; Chatterjee et al., 2017a; Tebbifakhr et al., 2018; Junczys-Dowmunt and Grundkiewicz, 2018). From the industry standpoint, APE has started to attract MT market players interested in combining the two technologies to support human translation in professional workflows (Crego et al., 2016).

Focusing on this industry-oriented perspective, this paper makes a step further on APE research by exploring an online neural approach to the task. The goal is to leverage human feedback (post edits) to improve on-the-fly a neural APE model without the need of stopping it for fine-tuning or re-training from scratch. Online learning capabilities are crucial (both for APE and MT) in computer-assisted translation scenarios where professional translators operate on suggestions provided by machines. In such scenarios, human corrections represent an invaluable source of knowledge that systems should exploit to enhance users' experience and increase their productivity.

Towards these objectives we provide two contributions. One is the first online approach to neural APE. Indeed, while MT-like online learning techniques have been proposed for phrase-based APE (Ortiz-Martínez and Casacuberta, 2014; Simard and Foster, 2013; Chatterjee et al., 2017b), nothing has been done yet under the state-of-the-art neural paradigm. In doing this, the other contribution is the first evaluation of neural APE run on the output of neural MT (NMT). So far, published results report significant gains[1] when APE is run to correct the output of a phrase-based MT system. To our knowledge, the true potential of APE with higher quality NMT output has not been investigated yet. The last observation introduces a more general discussion on the relation between MT and APE. Since, by definition, APE's reason of being is the sub-optimal quality of MT output, one might wonder if the level of current MT technology still justifies efforts on APE. Along this direction, our third contribution is an analysis of online neural APE applied to the output of NMT systems featuring different levels of performance. Our competitors range from a generic model trained on large parallel data (mimicking the typical scenario in which industry users – *e.g.* Language Service Providers – rely on web-based services or other black-box systems) to highly customized online models (like those that LSPs would desire but typically cannot afford). Our experiments in this range of conditions aim to shed light on the future of APE from the industry standpoint by answering the question: At what cost on the MT side will APE become useless?

2 Online neural APE

APE training data usually consist of (*src*, *mt*, *hpe*) triplets whose elements are: a source sentence (*src*), its translation (*mt*) and a human correction of the translated sentence (*hpe*). Models trained on such triplets are then used to correct the *mt* element of (*src*, *mt*) test data. Neural approaches to the task have shown their effectiveness in batch conditions, in which a static pre-trained model is run on the whole test corpus. When moving to an online setting, instead, APE systems should ideally be able to continuously evolve by stepwise learning from the interaction with the user. This means that, each time a new post-edit becomes available, the model has to update its parameters on-the-fly in order to produce better output for the next incoming sentence. To this aim, we extend a batch APE model by adding the capability to continuously learn from human corrections of its own output. This is done in two steps:
(1) *Before* post-editing, by means of an instance selection mechanism that updates the model by learning from previously collected triplets that are similar to the input test item (see lines 2-5 in Algorithm 1);
(2) *After* post-editing, by means of a model adaptation procedure that learns from human revisions of the last automatic correction generated by the system (lines 8-10).

Similar to the methods proposed in (Chatterjee et al., 2017b) and (Farajian et al., 2017), the instance-selection technique (first update step) consists of two components: *i)* a knowledge base (*KB*) that is continuously fed with the processed triplets, and *ii)* an information retrieval engine that, given the (*src*, *mt*) test item, selects the most similar triplet (lines 2-3). The engine is simultaneously queried using both *src* and *mt* segments and it returns the triplet that has the highest cosine similarity with both (*Top(R)*). If the similarity is above a threshold τ, a few training iterations are run to update the model parameters (line 5). Depending on the application scenario, *KB* can be pre-filled with the APE training data or left empty and filled only with the incoming triplets. In our experiments, the repository is initially empty.

```
Algorithm 1: Online neural APE
Require $M$: Trained APE model
Require $Ts$: Stream of test data
Require $KB$: Pool of ($src$, $mt$, $hpe$) triplets
1: while pop ($src$, $mt$) from $Ts$ do
2:     $R \leftarrow$ Retrieve (($src$, $mt$), $KB$)
3:     $(src_{top}, mt_{top}, hpe_{top}) \leftarrow$ Top ($R$)
4:     if Sim $((src_{top}, mt_{top}, hpe_{top}), (src, mt)) > \tau$ do
5:         $M^{*} \leftarrow$ Update $(M, (src_{top}, mt_{top}, hpe_{top}))$
6:     $ape \leftarrow$ APE $(M^{*}, (src, mt))$
7:     $hpe \leftarrow$ HumanPostEdit $((src, ape))$
8:     $KB \leftarrow KB \cup (src, mt, hpe)$
9:     $M^{**} \leftarrow$ Update $(M^{*}, (src, mt, hpe))$
10:    $M \leftarrow M^{**}$
11: end while
```

Once the *hpe* has been generated, the second update step takes place (line 9) by running few training iterations on the (*src*, *hpe*) pair. When training using only one single data point, the learning rate and the number of epochs have a crucial role because too high/small values can make the training unstable/inefficient. To avoid such problems, we connect the two parameters by applying a time-based decay learning rate that reduces the learning rate when increasing of the number of epochs (*i.e.* $lr = lr/(1+num\_epoch)$). In our experiments, this strategy results in better performance than setting a fixed learning rate.

[1] Up to 7.6 BLEU points at WMT 2017 (Bojar et al., 2017)

3 Experiments

We run our experiments on English-Italian data, by comparing the performance of different neural APE models (batch and online) used to correct the output of NMT systems of increasing quality.

3.1 Data

To train our NMT models we use both generic and in-domain data. Generic data cover a variety of domains. They comprise about 53M parallel sentences collected from publicly-available collections (*i.e.* all the English-Italian parallel corpora available on OPUS[2]) and about 50M sentence pairs from proprietary translation memories. Generic data, whose size is *per se* sufficient to train a competitive general-purpose engine, are used to build our basic NMT model. On top of it, in-domain (information technology) data are used in different ways to obtain improved, domain-adapted models. In-domain data are selected to emulate the online setting of industrial scenarios where input documents are processed sequentially on a sentence-by-sentence basis. They consist in a proprietary translation project of about 421K segments, which are split in training (416K segments) and test (5,472) keeping the sentence order. Post-edits are simulated using references.

To train the APE models we use the English-Italian section of the eSCAPE corpus (Negri et al., 2018). It consists of about 6.6M synthetically-created triplets in which the *mt* element is produced with phrase-based and neural MT systems.

3.2 NMT models

Our NMT models feature increasing levels of complexity, so to represent a range of conditions in which a user (say a Language Service Provider) has access to different resources in terms of MT technology and/or data for training and adaptation. Our systems, ranked in terms of complexity with respect to these two dimensions are:

Generic (G). This model is trained on the large (103M) multi-domain parallel corpus. It represents the situation in which our LSP entirely relies on an off-the-shelf, black-box MT engine that cannot be improved via domain adaptation.

Generic Online (GO). This model extends G with the capability to learn from the incoming human post-edits (5,472 test items). Before and after translation, few training iterations adapt it to the domain of the input document. The adaptation steps implement the same strategies of the online APE system (see §2). This setting represents the situation in which our LSP has access to the inner workings of a competitive online NMT system.

Specialized (S). This model is built by fine-tuning (Luong and Manning, 2015) G on the in-domain training data (416K). It reflects the condition in which our LSP has access both to customer's data and to the inner workings of a competitive *batch* NMT engine. The adaptation routine, however, is limited to the standard approach of performing additional training steps on the in-domain data.

Specialized Online (SO). This model is built by combining the functionalities of GO and S. It uses the in-domain training data for fine-tuning and the incoming (*src*, *hpe*) pairs for online adaptation to the target domain. This setting represents the situation in which our LSP has access to: *i)* customer's in-domain data and *ii)* the inner workings of a competitive *online* NMT engine.

All the models are trained with the ModernMT open source software,[3] which is built on top of OpenNMT-py (Klein et al., 2017). It employs an LSTM-based recurrent architecture with attention (Bahdanau et al., 2014) using 2 bi-directional LSTM layers in the encoder, 4 left-to-right LSTM layers in the decoder, and a dot-product attention model (Luong et al., 2015). In our experiments we used an embeddings' size of 1024, LSTMs of size 1024, and a source and target vocabulary of 32K words, jointly trained with the BPE algorithm (Sennrich et al., 2016). The fact that ModernMT already implements the online adaptation method presented in (Farajian et al., 2017) simplified our tests with online neural APE run on the output of competitive NMT systems (GO and SO).

3.3 APE models

We experiment with two neural APE systems:

Generic APE. This batch system is trained only on generic data (6.6M triplets from eSCAPE) and is similar to those tested in the APE shared task at WMT. The main difference is that the training data are neither merged with in-domain triplets nor selected based on target domain information.

Online APE. This system is trained on the generic data and continuously learns from human post-edits of the test set as described in §2.

[2] `http://opus.lingfil.uu.se` dump of mid June 2017.

[3] `http://github.com/ModernMT/MMT`.

MT Type	MT	Generic APE	Online APE
Generic (G)	40.3	39.0	**47.1**†
Gen. Online (GO)	45.6	41.9	**48.1**†
Specialized (S)	52.1	45.5	**53.5**†
Spec. Online (SO)	**55.0**	47.4	54.8

Table 1: APE performance on NMT outputs of different quality ("†" denotes statistically significant differences wrt. the MT baseline with $p<0.05$).

The two systems are based on a multi-source attention-based encoder-decoder approach similar to (Chatterjee et al., 2017a). It employs a GRU-based recurrent architecture with attention and uses two independent encoders to process the *src* and *mt* segments. Similar to the NMT systems, it is trained on sub-word units by using BPE, with a vocabulary created by selecting to 50K most frequent sub-words. Word embedding and GRU hidden state sizes are set to 1024. Network parameters are optimized with Adagrad (Duchi et al., 2011) with a learning rate of 0.01. A development set randomly extracted from the training data is used to set the similarity threshold used by the online model for the first update step (τ=0.5) as well as the learning rate (0.01) and the number of epochs (3) of both adaptation steps.

4 Results and discussion

APE results computed on different levels of translation quality are reported in Table 1. Looking at the NMT performance, all the adaptation techniques yield significant improvements over the Generic model (G). The large gain achieved via fine-tuning on in-domain data (S: +11.8 BLEU) is further increased when adding online learning capabilities on top of it to create the most competitive Specialized Online system (SO: +14.7).

As expected, the batch APE model trained on generic data only (that is, without in-domain information) is unable to improve the quality of raw MT output. Moreover, although APE results increase with higher translation quality, also the performance distance from the more competitive NMT systems becomes larger (from -1.3 to -7.6 points respectively for G and SO). These results confirm the WMT findings about the importance of domain customization for batch APE (Bojar et al., 2017), and advocate for online solutions capable to maximize knowledge exploitation at test time by learning from user feedback.

Online APE achieves significant[4] improvements not only over the output of G (+6.8) and its online extension GO (+2.5), but also over the specialized model S (+1.4). The gain over GO is particularly interesting: it shows that even when APE and MT use the same in-domain data for online adaptation, the APE model is more reactive to human feedback. Though trained on much smaller generic corpora (6.6M triplets versus 103M parallel sentences), the possibility to leverage richer information in the form of (*src*, *mt*, *pe*) instances at test time seems to have a positive impact. A deeper exploration of this aspect falls out of the scope of this paper and is left as future work.

Also with online APE, the gains become smaller by increasing the MT quality, reaching a point where the system can only approach the highest MT performance of SO (with a non-significant -0.2 BLEU difference). This confirms that correcting the output of competitive NMT engines is a hard task, even for a dynamic APE system that learns from the interaction with the user. However, besides improving its performance by learning from user feedback acquired at test time (similar to the APE system), SO also relies on previous fine-tuning on a large in-domain corpus (similar to S). To answer our initial question ("*At what cost on the MT side will APE become useless?*") it is worth remarking that leveraging in-domain training/adaptation data is a considerable advantage for MT but it comes at a cost that should not be underestimated. In terms of the data itself, collecting enough parallel sentences for each target domain is a considerable bottleneck that limits the scalability of competitive NMT solutions. In addition to that, the technology requirements (*i.e.* having access to the inner workings of the NMT engine) and the computational costs involved (for fine-tuning the generic model) are constraints that few LSPs are probably able to satisfy.

5 Conclusion

We introduced an online neural APE system, which is trained on generic data and only exploits user feedback to improve its performance, and evaluated it on the output of NMT systems featuring increasing complexity and in-domain data demand. Our results show the effectiveness of current APE technology in the typical setting of

[4] Statistical significance is computed with paired bootstrap resampling (Koehn, 2004).

most LSPs while, in terms of resources (especially in-domain data) and technical expertise needed. We also conclude that developing MT engines that make APE useless is still a prerogative of few.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *arXiv preprint arXiv:1409.0473*.

Hanna Béchara, Yanjun Ma, and Josef van Genabith. 2011. Statistical Post-Editing for a Statistical MT System. In *Proceedings of the 13th Machine Translation Summit*, pages 308–315, Xiamen, China, September.

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, September.

Rajen Chatterjee, Marion Weller, Matteo Negri, and Marco Turchi. 2015. Exploring the Planet of the APEs: a Comparative Study of State-of-the-art Methods for MT Automatic Post-Editing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics)*, pages 156–161, Beijing, China, July.

Rajen Chatterjee, M. Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017a. 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, Volume 2: Shared Task Papers*, pages 630–638, Copenhagen, Denmark, September.

Rajen Chatterjee, Gebremedhen Gebremelak, Matteo Negri, and Marco Turchi. 2017b. Online Automatic Post-editing for MT in a Multi-Domain Translation Environment. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 525–535, Valencia, Spain, April.

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*, Brussels, Belgium, October. Association for Computational Linguistics.

Josep Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, et al. 2016. SYSTRAN's Pure Neural Machine Translation Systems. *arXiv preprint arXiv:1610.05540*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159, July.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-Domain Neural Machine Translation through Unsupervised Adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark, September.

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*, pages 751–758, Berlin, Germany, August.

Marcin Junczys-Dowmunt and Roman Grundkiewicz. 2018. Microsoft and University of Edinburgh at WMT2018: Dual-Source Transformer for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.

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, July.

Kevin Knight and Ishwar Chander. 1994. Automated Post-Editing of Documents. In *Proceedings of AAAI*, volume 94, pages 779–784.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Empirical Methods on Natural Language Processing*, pages 388–395, Barcelona, Spain, July.

Antonio L. Lagarda, Daniel Ortiz-Martínez, Vicent Alabau, and Francisco Casacuberta. 2015. Translating without In-domain Corpus: Machine Translation Post-Editing with Online Learning Techniques. *Computer Speech & Language*, 32(1):109–134.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford Neural Machine Translation Systems for Spoken Language Domains. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT'15)*, pages 76–79, Da Nang, Vietnam, December.

Minh Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *arXiv preprint arXiv:1508.04025*.

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, May.

Daniel Ortiz-Martínez and Francisco Casacuberta. 2014. The New THOT Toolkit for Fully-Automatic and Interactive Statistical Machine Translation. In *Proceedings of the 14th Annual Meeting of the European Association for Computational Linguistics*, pages 45–48, Gothenburg, Sweden, April.

Rudolf Rosa, David Marecek, and Ales Tamchyna. 2013. Deepfix: Statistical Post-editing of Statistical Machine Translation Using Deep Syntactic Analysis. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 172–179, Sofia, Bulgaria, August.

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, August.

Michel Simard and George Foster. 2013. PEPr: Post-edit Propagation Using Phrase-based Statistical Machine Translation. In *Proceedings of the XIV Machine Translation Summit*, pages 191–198, Nice, France, September.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-Based Post-Editing. In *Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 508–515, Rochester, New York, April.

Amirhossein Tebbifakhr, Ruchit Agrawal, Rajen Chatterjee, Matteo Negri, and Marco Turchi. 2018. Multi-source Transformer with Combined Losses for Automatic Post-Editing. In *Proceedings of the Third Conference on Machine Translation*, Brussels, Belgium, October.