A Shared Attention Mechanism for Interpretation of Neural Automatic Post-Editing Systems

Inigo Jauregi Unanue[1,2], **Ehsan Zare Borzeshi**[2], **Massimo Piccardi**[1]
[1] University of Technology Sydney, Sydney, Australia
[2] Capital Markets Cooperative Research Centre, Sydney, Australia

{ijauregi,ezborzeshi}@cmcrc.com, massimo.piccardi@uts.edu.au

Abstract

Automatic post-editing (APE) systems aim to correct the systematic errors made by machine translators. In this paper, we propose a neural APE system that encodes the source (*src*) and machine translated (*mt*) sentences with two separate encoders, but leverages a shared attention mechanism to better understand how the two inputs contribute to the generation of the post-edited (*pe*) sentences. Our empirical observations have showed that when the *mt* is incorrect, the attention shifts weight toward tokens in the *src* sentence to properly edit the incorrect translation. The model has been trained and evaluated on the official data from the WMT16 and WMT17 APE IT domain English-German shared tasks. Additionally, we have used the extra 500K artificial data provided by the shared task. Our system has been able to reproduce the accuracies of systems trained with the same data, while at the same time providing better interpretability.

1 Introduction

In current professional practice, translators tend to follow a two-step approach: first, they run a machine translator (MT) to obtain a first-cut translation; then, they manually correct the MT output to produce a result of adequate quality. The latter step is commonly known as *post-editing* (PE). Stemming from this two-step approach and the recent success of deep networks in MT (Sutskever et al., 2014; Bahdanau et al., 2014; Luong et al., 2015), the MT research community has devoted increasing attention to the task of automatic post-editing (APE) (Bojar et al., 2017).

The rationale of an APE system is to be able to automatically correct the systematic errors made by the MT and thus dispense with or reduce the work of the human post-editors. The data for training and evaluating these systems usually consist of triplets (*src*, *mt*, *pe*), where *src* is the sentence in the source language, *mt* is the output of the MT, and *pe* is the human post-edited sentence. Note that the *pe* is obtained by correcting the *mt*, and therefore these two sentences are closely related. An APE system is "monolingual" if it only uses the *mt* to predict the post-edits, or "contextual" if it uses both the *src* and the *mt* as inputs (Béchara et al., 2011).

Despite their remarkable progress in recent years, neural APE systems are still elusive when it comes to interpretability. In deep learning, highly interpretable models can help researchers to overcome outstanding issues such as learning from fewer annotations, learning with human-computer interactions and debugging network representations (Zhang and Zhu, 2018). More specifically in APE, a system that provides insights on its decisions can help the human post-editor to understand the system's errors and consequently provide better corrections. As our main contribution, in this paper we propose a contextual APE system based on the seq2seq model with attention which allows for inspecting the role of the *src* and the *mt* in the editing. We modify the basic model with two separate encoders for the *src* and the *mt*, but with a single attention mechanism shared by the hidden vectors of both encoders. At each decoding step, the shared attention has to decide whether to place more weight on the tokens from the *src* or the *mt*. In our experiments, we clearly observe that when the *mt* translation contains mistakes (word order, incorrect words), the model learns to shift the attention toward tokens in the source language, aiming to get extra "context" or information that will help to correctly edit the translation. Instead, if

Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pages 11–17
Melbourne, Australia, July 20, 2018.

the *mt* sentence is correct, the model simply learns to pass it on word by word. In Section 4.4, we have plotted the attention weight matrices of several predictions to visualize this finding.

The model has been trained and evaluated with the official datasets from the WMT16 and WMT17 Information Technology (IT) domain APE English-German (en-de) shared tasks (Bojar et al., 2016, 2017). We have also used the 500K artificial data provided in the shared task for extra training. For some of the predictions in the test set, we have analysed the plots of attention weight matrices to shed light on whether the model relies more on the *src* or the *mt* at each time step. Moreover, our model has achieved higher accuracy than previous systems that used the same training setting (official datasets + 500K extra artificial data).

2 Related work

In an early work, (Simard et al., 2007) combined a rule-based MT (RBMT) with a statistical MT (SMT) for monolingual post-editing. The reported results outperformed both systems in standalone translation mode. In 2011, (Béchara et al., 2011) proposed the first model based on contextual post-editing, showing improvements over monolingual approaches.

More recently, neural APE systems have attracted much attention. (Junczys-Dowmunt and Grundkiewicz, 2016) (the winner of the WMT16 shared task) integrated various neural machine translation (NMT) components in a log-linear model. Moreover, they suggested creating artificial triplets from out-of-domain data to enlarge the training data, which led to a drastic improvement in PE accuracy. Assuming that post-editing is reversible, (Pal et al., 2017) have proposed an attention mechanism over bidirectional models, *mt*→ *pe* and *pe* → *mt*. Several other researchers have proposed using multi-input seq2seq models for contextual APE (Bérard et al., 2017; Libovický et al., 2016; Varis and Bojar, 2017; Pal et al., 2017; Libovický and Helcl, 2017; Chatterjee et al., 2017). All these systems employ separate encoders for the two inputs, *src* and *mt*.

2.1 Attention mechanisms for APE

A key aspect of neural APE systems is the attention mechanism. A conventional attention mechanism for NMT first learns the alignment scores (e^{ij}) with an alignment model (Bahdanau et al., 2014; Luong et al., 2015) given the j-th hidden vector of the encoder ($\mathbf{h}^j$) and the decoder's hidden state ($\mathbf{s}_{i-1}$) at time $i-1$ (Equation 1). Then, Equation 2 computes the normalized attention weights, with T_x the length of the input sentence. Finally, the context vector is computed as the sum of the encoder's hidden vectors weighed by the attention weights (Equation 3). The decoder uses the computed context vector to predict the output.

$$e^{ij} = aligment\_model(\mathbf{h}^j, \mathbf{s}^{i-1}) \quad (1)$$

$$\alpha^{ij} = \frac{exp(e^{ij})}{\sum_{m=1}^{T_x} exp(e^{im})} \quad (2)$$

$$\mathbf{c}^i = \sum_{j=1}^{T_x} \alpha^{i,j} \mathbf{h}^j \quad (3)$$

In the APE literature, two recent papers have extended the attention mechanism to contextual APE. (Chatterjee et al., 2017) (the winner of the WMT17 shared task) have proposed a two-encoder system with a separate attention for each encoder. The two attention networks create a context vector for each input, $\mathbf{c}_{src}$ and $\mathbf{c}_{mt}$, and concatenate them using additional, learnable parameters, $\boldsymbol{W}_{ct}$ and $\mathbf{b}_{ct}$, into a merged context vector, $\mathbf{c}_{merge}$ (Equation 4).

$$\mathbf{c}^i_{merge} = [\mathbf{c}^i_{src}; \mathbf{c}^i_{mt}] * \boldsymbol{W}_{ct} + \mathbf{b}_{ct} \quad (4)$$

(Libovický and Helcl, 2017) have proposed, among others, an attention strategy named the *flat attention*. In this approach, all the attention weights corresponding to the tokens in the two inputs are computed with a joint soft-max:

$$\alpha^{ij}_{(k)} = \frac{exp(e^{ij}_{(k)})}{\sum_{n=1}^{2} \sum_{m=1}^{T_x^{(n)}} exp(e^{im}_{(n)})} \quad (5)$$

where $e^{ij}_{(k)}$ is the attention energy of the j-th step of the k-th encoder at the i-th decoding step and $T_x^{(k)}$ is the length of the input sequence of the k-th encoder. Note that because the attention weights are computed jointly over the different encoders, this approach allows observing whether the system assigns more weight to the tokens of the *src* or the *mt* at each decoding step. Once the attention weigths are computed, a single context vector ($\mathbf{c}$) is created as:

$$\mathbf{c}^i = \sum_{k=1}^{N} \sum_{j=1}^{T_x^{(k)}} \alpha_{(k)}^{i,j} \mathbf{U}_{c(k)} \mathbf{h}_{(k)}^j \quad (6)$$

where $\mathbf{h}_{(k)}^j$ is the j-th hidden vector from the k-th encoder, $T_x^{(k)}$ is the number of hidden vectors from the k-th encoder, and $\mathbf{U}_{c(k)}$ is the projection matrix for the k-th encoder that projects its hidden vectors to a common-dimensional space. This parameter is also learnable and can further re-weigh the two inputs.

3 The proposed model

The main focus of our paper is on the interpretability of the predictions made by neural APE systems. To this aim, we have assembled a contextual neural model that leverages two encoders and a shared attention mechanism, similarly to the *flat attention* of (Libovický and Helcl, 2017). To describe it, let us assume that $\mathbf{X}_{src} = \{\mathbf{x}_{src}^1, ..., \mathbf{x}_{src}^N\}$ is the *src* sentence and $\mathbf{X}_{mt} = \{\mathbf{x}_{mt}^1, ..., \mathbf{x}_{mt}^M\}$ is the *mt* sentence, where N and M are their respective numbers of tokens. The two encoders encode the two inputs separately:

$$\begin{aligned} \mathbf{h}_{src}^j &= enc_{src}(\mathbf{x}_{src}^j, \mathbf{h}_{src}^{j-1}) \quad j = 1, ..., N \\ \mathbf{h}_{mt}^j &= enc_{mt}(\mathbf{x}_{mt}^j, \mathbf{h}_{mt}^{j-1}) \quad j = 1, ..., M \end{aligned} \quad (7)$$

All the hidden vectors outputs by the two encoders are then concatenated as if they were coming from a single encoder:

$$\mathbf{h}_{join} = \{\mathbf{h}_{src}^1, ..., \mathbf{h}_{src}^N, \mathbf{h}_{mt}^1, ..., \mathbf{h}_{mt}^M\} \quad (8)$$

Then, the attention weights and the context vector at each decoding step are computed from the hidden vectors of $\mathbf{h}_{join}$ (Equations 9-11):

$$e^{ij} = aligment\_model(\mathbf{h}_{join}^j, \mathbf{s}^{i-1}) \quad (9)$$

$$\alpha^{ij} = \frac{exp(e^{ij})}{\sum_{m=1}^{N+M} exp(e^{im})} \quad (10)$$

$$\mathbf{c}^i = \sum_{j=1}^{N+M} \alpha^{i,j} \mathbf{h}_{join}^j \quad (11)$$

where i is the time step on the decoder side, j is the index of the hidden encoded vector. Given that the $\alpha^{i,j}$ weights form a normalized probability distribution over j, this model is "forced" to spread the weight between the *src* and *mt* inputs. Note that our model differs from that proposed by (Libovický and Helcl, 2017) only in that we do not employ the learnable projection matrices, $U_{c(k)}$. This is done to avoid re-weighing the contribution of the two inputs in the context vectors and, ultimately, in the predictions. More details of the proposed model and its hyper-parameters are provided in Section 4.3.

4 Experiments

4.1 Datasets

For training and evaluation we have used the WMT17 APE[1] IT domain English-German dataset. This dataset consists of 11,000 triplets for training, 1,000 for validation and 2,000 for testing. The hyper-parameters have been selected using only the validation set and used unchanged on the test set. We have also trained the model with the 12,000 sentences from the previous year (WMT16), for a total of 23,000 training triplets.

4.2 Artificial data

Since the training set provided by the shared task is too small to effectively train neural networks, (Junczys-Dowmunt and Grundkiewicz, 2016) have proposed a method for creating extra, "artificial" training data using round-trip translations. First, a language model of the target language (German here) is learned using a monolingual dataset. Then, only the sentences from the monolingual dataset that have low perplexity are round-trip translated using two off-the-shelf translators (German-English and English-German). The low-perplexity sentences from the monolingual dataset are treated as the *pe*, the German-English translations as the *src*, and the English-German back-translations as the *mt*. Finally, the (*src*, *mt*, *pe*) triplets are filtered to only retain sentences with comparable TER statistics to those of the manually-annotated training data. These artificial data have proved very useful for improving the accuracy of several neural APE systems, and they have therefore been included in the WMT17 APE shared task. In this paper, we have limited ourselves to using 500K artificial triplets as done in (Varis and Bojar, 2017; Bérard

[1] http://www.statmt.org/wmt17/ape-task.html

# encoders	2
encoder type	B-LSTM
encoder layers	2
encoder hidden dim	500
# decoders	1
decoder type	LSTM
decoder layers	2
decoder hidden dim	500
word vector dim	300
attention type	*general*
dropout	0.3
beam size	5

Table 1: The model and its hyper-parameters.

et al., 2017). To balance artificial and manually-annotated data during training, we have resampled the official 23K triplets 10 times.

4.3 Training and hyper-parameters

Hereafter we provide more information about the model's implementation, its hyper-parameters, the pre-processing and the training to facilitate the reproducibility of our results. We have made our code publicly available[2].

To implement the encoder/decoder with separate encoders for the two inputs (*src*, *mt*) and a single attention mechanism, we have modified the open-source OpenNMT code (Klein et al., 2017).

Table 1 lists all hyper-parameters which have all been chosen using only training and validation data. The two encoders have been implemented using a Bidirectional Long Short-Term Memory (B-LSTM) (Hochreiter and Schmidhuber, 1997) while the decoder uses a unidirectional LSTM. Both the encoders and the decoder use two hidden layers. For the attention network, we have used the OpenNMT's *general* option (Luong et al., 2015).

As for the pre-processing, the datasets come already tokenized. Given that German is a morphologically rich language, we have learned the subword units using the BPE algorithm (Sennrich et al., 2015) only over the official training sets from the WMT16 and WMT17 IT-domain APE shared task (23,000 sentences). The number of *merge* operations has been set to 30,000 under the intuition that one or two word splits per sentence could suffice. Three separate vocabularies have been used for the (*src*, *mt* and *pe*) sentences. Each vocabulary contains a maximum of 50,000 most-frequent subword units; the remaining tokens are treated as unknown (<*unk*>).

[2]https://github.com/ijauregiCMCRC/Shared_Attention_for_APE

Model	TER	BLEU
MT (Bojar et al., 2017)	24.48	62.49
SPE (Bojar et al., 2017)	24.69	62.97
(Varis and Bojar, 2017)	24.03	64.28
(Bérard et al., 2017)	22.81	65.91
train 11K	41.58	43.05
train 23K	30.23	57.14
train 23K + 500K	**22.60**	**66.21**

Table 2: Results on the WMT17 IT domain English-German APE test set.

As mentioned in Section 4.2, we have trained our model with 500K extra triplets as in (Bérard et al., 2017). We have oversampled the 23K official triplets 10 times, added the extra 500K, and trained the model for 20 epochs. We have used Stochastic Gradien Descent (SGD) with a learning rate of 1 and a learning rate decay of 0.5. The learning rate decays if there are no improvements on the validation set.

In all cases, we have selected the models and hyper-parameters that have obtained the best results on the validation set (1,000 sentences), and reported the results blindly over the test set (2,000 sentences). The performance has been evaluated in two ways: first, as common for this task, we have reported the accuracy in terms of Translation Error Rate (TER) (Snover et al., 2006) and BLEU score (Papineni et al., 2002). Second, we present an empirical analysis of the attention weight matrices for some notable cases.

4.4 Results

Table 2 compares the accuracy of our model on the test data with two baselines and two state-of-the-art comparable systems. The MT baseline simply consists of the accuracy of the *mt* sentences with respect to the *pe* ground truth. The other baseline is given by a statistical PE (SPE) system (Simard et al., 2007) chosen by the WMT17 organizers. Table 2 shows that when our model is trained with only the 11K WMT17 official training sentences, it cannot even approach the baselines. Even when the 12K WMT16 sentences are added, its accuracy is still well below that of the baselines. However, when the 500K artificial data are added, it reports a major improvement and it outperforms them both significantly. In addition, we have compared our model with two recent systems that have used our

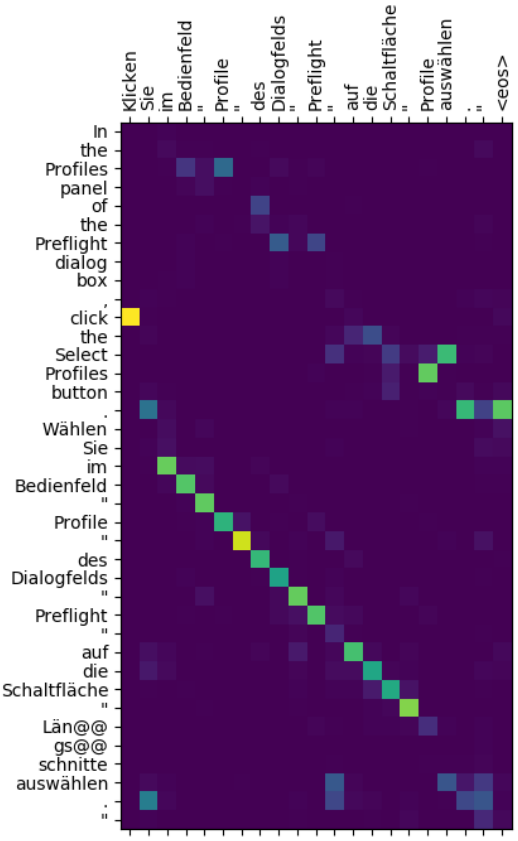

Figure 1: An example of perfect correction of an *mt* sentence.

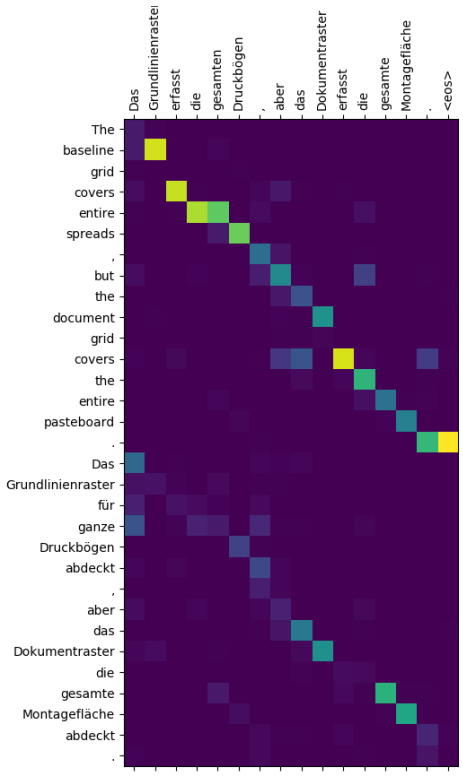

Figure 2: Partial improvement of an *mt* sentence.

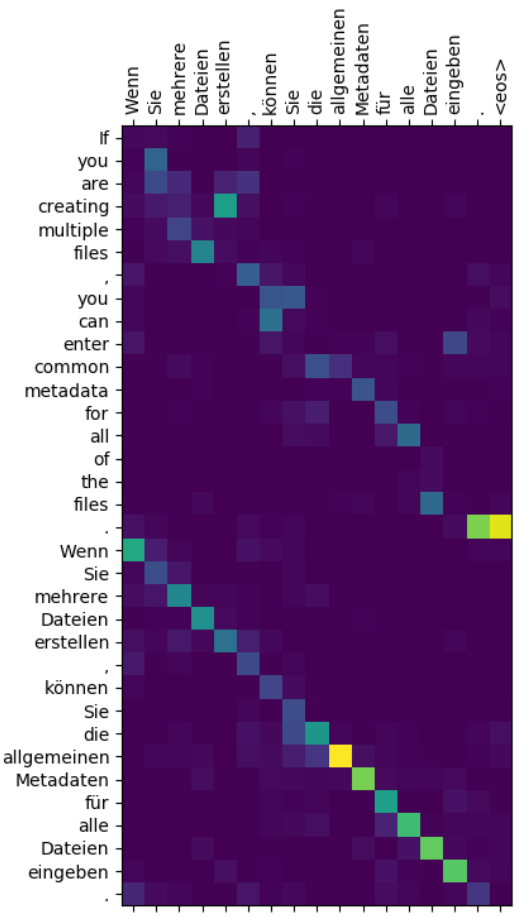

Figure 3: Passing on a correct *mt* sentence.

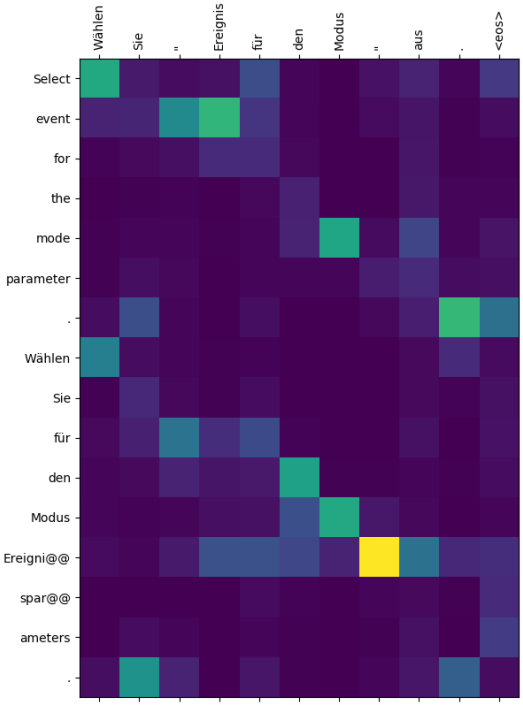

Figure 4: A completely incorrect prediction.

same training settings (500K artificial triplets + 23K manual triplets oversampled 10 times), reporting a slightly higher accuracy than both (1.43 TER and 1.93 BLEU p.p. over (Varis and Bojar, 2017) and 0.21 TER and 0.30 BLEU p.p. over (Bérard et al., 2017)). Since their models explicitly predicts edit operations rather than post-edited sentences, we speculate that these two tasks are of comparable intrinsic complexity.

In addition to experimenting with the proposed model (Equation 11), we have also tried to add the projection matrices of the flat attention of (Libovický and Helcl, 2017) (Equation 6). However, the model with these extra parameters showed evident over-fitting, with a lower perplexity on the training set, but unfortunately also a lower BLEU score of 53.59 on the test set. On the other hand, (Chatterjee et al., 2017) and other participants of the WMT 17 APE shared task [3] were able to achieve higher accuracies by using 4 million artificial training triplets. Unfortunately, using such a large dataset sent the computation out of memory on a system with 32 GB of RAM. Nonetheless, our main goal is not to establish the highest possible accuracy, but rather contribute to the interpretability of APE predictions while reproducing approximately the same accuracy of current systems trained in a comparable way.

For the analysis of the interpretability of the system, we have plotted the attention weight matrices for a selection of cases from the test set. These plots aim to show how the shared attention mechanism shifts the attention weights between the tokens of the *src* and *mt* inputs at each decoding step. In the matrices, the rows are the concatenation of the *src* and *mt* sentences, while the columns are the predicted *pe* sentence. To avoid cluttering, the ground-truth *pe* sentences are not shown in the plots, but they are commented upon in the discussion. Figure 1 shows an example where the *mt* sentence is almost correct. In this example, the attention focuses on passing on the correct part. However, the start (*Wählen*) and end (*Längsschnitte*) of the *mt* sentence are wrong: for these tokens, the model learns to place more weight on the English sentence (*click* and *Select Profiles*). The predicted *pe* is eventually identical to the ground truth.

Conversely, Figure 2 shows an example where the *mt* sentence is rather incorrect. In this case, the model learns to focus almost completely on the English sentence, and the prediction is very aligned with it. The predicted *pe* is not identical to the ground truth, but it is significantly more accurate than the *mt*. Figure 3 shows a case of a perfect *mt* translation where the model simply learns to pass the sentence on word by word. Eventually, Figure 4 shows an example of a largely incorrect *mt* where the model has not been able to properly edit the translation. In this case, the attention matrix is scattered and defocused.

[3] http://www.statmt.org/wmt17/ape-task.html

Sentence	Focus
src	23%
mt	45%
Both	31%

Table 3: Percentage of the decoding steps with marked attention weight on either input (*src*, *mt*) or both.

In addition to the visualizations of the attention weights, we have computed an attention statistic over the test set to quantify the proportions of the two inputs. At each decoding time step, we have added up the attention weights corresponding to the *src* input ($\alpha^i_{src} = \sum_{j=1}^{N} \alpha^{ij}$) and those corresponding to the *mt* ($\alpha^i_{mt} = \sum_{j=N+1}^{N+M} \alpha^{ij}$). Note that, obviously, $\alpha^i_{src} + \alpha^i_{mt} = 1$. Then, we have set an arbitrary threshold, $t = 0.6$, and counted step i to the *src* input if $\alpha^i_{src} > t$. If instead $\alpha^i_{src} < 1 - t$, we counted the step to the *mt* input. Eventually, if $1 - t \leq \alpha^i_{src} \leq t$, we counted the step to both inputs. Table 3 shows this statistic. Overall, we have recorded 23% decoding steps for the *src*, 45% for the *mt* and 31% for both. It is to be expected that the majority of the decoding steps would focus on the *mt* input if it is of sufficient quality. However, the percentage of focus on the *src* input is significant, confirming its usefulness.

5 Conclusion

In this paper, we have presented a neural APE system based on two separate encoders that share a single, joined attention mechanism. The shared attention has proved a key feature for inspecting how the selection shifts on either input, (*src* and *mt*), at each decoding step and, in turn, understanding which inputs drive the predictions. In addition to its easy interpretability, our model has reported a competitive accuracy compared to recent, similar systems (i.e., systems trained with the official WMT16 and WMT17 data and 500K extra

training triplets). As future work, we plan to continue to explore the interpretability of contemporary neural APE architectures.

6 Acknowledgements

We would like to acknowledge the financial support received from the Capital Markets Cooperative Research Centre (CMCRC), an applied research initiative of the Australian Government.

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 *MT Summit*. volume 13, pages 308–315.

Olivier Bérard, Alexandre Pietquin, and Laurent Besacier. 2017. Lig-cristal system for the wmt17 automatic post-editing task. In *Proceedings of the Second Conference on Machine Translation*. pages 623–629.

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*. pages 169–214.

Ondřej Bojar, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, Varvara Logacheva, Christof Monz, Matteo Negri, Aurélie Névéol, 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: Volume 2, Shared Task Papers*. volume 2, pages 131–198.

Rajen Chatterjee, M Amin Farajian, Matteo Negri, Marco Turchi, Ankit Srivastava, and Santanu Pal. 2017. Multi-source neural automatic post-editing: Fbks participation in the wmt 2017 ape shared task. In *Proceedings of the Second Conference on Machine Translation*. pages 630–638.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

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 751–558.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810* .

Jindřich Libovickỳ and Jindřich Helcl. 2017. Attention strategies for multi-source sequence-to-sequence learning. *arXiv preprint arXiv:1704.06567* .

Jindřich Libovickỳ, Jindřich Helcl, Marek Tlustỳ, Pavel Pecina, and Ondřej Bojar. 2016. Cuni system for wmt16 automatic post-editing and multimodal translation tasks. In *Proceedings of the First Conference on Machine Translation*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025* .

Santanu Pal, Sudip Kumar Naskar, Mihaela Vela, Qun Liu, and Josef van Genabith. 2017. Neural automatic post-editing using prior alignment and reranking. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*. volume 2, pages 349–355.

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*. Association for Computational Linguistics, pages 311–318.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* .

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical phrase-based post-editing. In *NAACL HLT*. pages 505–515.

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*. volume 200.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*. pages 3104–3112.

Dusan Varis and Ondřej Bojar. 2017. Cuni system for wmt17 automatic post-editing task. In *Proceedings of the Second Conference on Machine Translation*. pages 661–666.

Quan-shi Zhang and Song-Chun Zhu. 2018. Visual interpretability for deep learning: a survey. *Frontiers of Information Technology & Electronic Engineering* 19(1):27–39.