Ensembling Factored Neural Machine Translation Models for Automatic Post-Editing and Quality Estimation

Chris Hokamp
ADAPT Centre
Dublin City University
chokamp@computing.dcu.ie

Abstract

This work presents a novel approach to Automatic Post-Editing (APE) and Word-Level Quality Estimation (QE) using ensembles of specialized Neural Machine Translation (NMT) systems. Word-level features that have proven effective for QE are included as input factors, expanding the representation of the original source and the machine translation hypothesis, which are used to generate an automatically post-edited hypothesis. We train a suite of NMT models that use different input representations, but share the same output space. These models are then ensembled together, and tuned for both the APE and the QE task. We thus attempt to connect the state-of-the-art approaches to APE and QE within a single framework. Our models achieve state-of-the-art results in both tasks, with the only difference in the tuning step which learns weights for each component of the ensemble.

1 Introduction

Translation destined for human consumption often must pass through multiple editing stages. In one common scenario, human translators correct machine translation (MT) output, correcting errors and omissions until a perfect translation has been produced. Several studies has shown that this process, referred to as "post-editing", is faster than translation from scratch (Specia, 2011), or interactive machine translation (Green et al., 2013).

A relatively recent line of research has tried to build models which correct errors in MT automatically (Simard et al., 2007; Bojar et al., 2015; Junczys-Dowmunt and Grundkiewicz, 2016). Automatic Post-Editing (APE) typically views the system that produced the original translation as a black box, which cannot be modified or inspected. An APE system has access to the same data that a human translator would see: a source sentence and a translation hypothesis. The job of the system is to output a corrected hypothesis, attempting to fix errors made by the original translation system. This can be viewed as a sequence-to-sequence task (Sutskever et al., 2014), and is also similar to multi-source machine translation (Zoph and Knight, 2016; Firat et al., 2016). However, APE intuitively tries to make the minimum number of edits required to transform the hypothesis into a satisfactory translation, because we would like our system to mimic human translators in attempting to minimize the time spent correcting each MT output. This additional constraint on APE models differentiates the task from multi-source MT.

The Word Level QE task is ostensibly a simpler version of APE, where a system must only decide whether or not each word in an MT hypothesis belongs in the post-edited version – it is not necessary to propose a fix for errors. Most recent work has considered word-level QE to be a sequence labeling task, and employed the standard tools of structured prediction to solve it, i.e. structured predictors such as CRFs or structured SVMs, which take advantage of sparse representations and very large feature sets, as well as dependencies between labels in the output sequence (Logacheva et al., 2016; Martins et al., 2016). However, Martins et al. (2017) recently proposed a new method of word-level QE using APE, which simply uses an APE system to produce a "pseudo-post-edit" given a source sentence and an MT hypothesis. Their approach, which we call **APE-QE**, is the basis of the work presented here. In APE-QE, the original MT hypothesis is then aligned with the pseudo-post-edit from the APE system using word level edit-

Proceedings of the Conference on Machine Translation (WMT), Volume 2: Shared Task Papers, pages 647–654
Copenhagen, Denmark, September 711, 2017.

distance, and words which correspond to *Insert* or *Delete* operations are labeled as incorrect. Note that this also corresponds exactly to the way QE datasets are currently created, with the only difference being that human post-edits are typically used to create gold-standard data (Bojar et al., 2015).

A key similarity between the QE and APE tasks is that both use information from two sequences: (1) the original source input, and (2) an MT hypothesis. Martins et al. (2017), showed that APE systems with no knowledge about the QE task already provide a very strong baseline for QE. Because the essential training data for the APE and QE tasks is identical, consisting of parallel triples of (SRC, MT, PE), it is also natural to consider these tasks as two subtasks that make use of a single underlying model.

In this work, we explicitly design ensembles of NMT models for both word-level QE, and APE. This approach builds upon the approach presented in Martins et al. (2017), by incorporating features which have proven effective for Word Level QE as "factors" in the input to Neural Machine Translation (NMT) systems. We achieve state-of-the-art results in both Automatic Post-Editing and Word-Level Quality Estimation, matching the performance of much more complex QE systems, and significantly outperforming the current state-of-the-art in APE.

The main contributions of this work are:

- Novel Input Representations for Neural APE models
- New tuned ensembles for APE-QE
- An open-source decoder supporting ensembles of models with different inputs[1]

The following sections discuss our approach to creating hybrid models for APE-QE, which should be able to solve both tasks with minimal modification.

2 Related Work

Two important lines of research have recently made breakthroughs in QE and APE.

[1]code avaiable at https://github.com/chrishokamp/constrained_decoding

2.1 Automatic Post-Editing

APE and QE training datasets consist of (SRC, MT, PE) triples, where the post-edited reference is created by a human translator in the workflow described above. However, publicly available APE datasets are relatively small in comparison to parallel datasets used to train machine translation systems. Junczys-Dowmunt and Grundkiewicz (2016) introduce a method for generating a large synthetic training dataset from a parallel corpus of (SRC, REF) by first translating the reference to the source language, and then translating this "pseudo-source" back into the target language, resulting in a “pseudo-hypothesis" which is likely to be more similar to the reference than a direct translation from source to target. The release of this synthetic training data was a major contribution towards improving APE.

Junczys-Dowmunt and Grundkiewicz (2016) also present a framework for ensembling SRC $\rightarrow$ PE and SRC $\rightarrow$ PE NMT models together, and tuning for APE performance. Our work extends this idea with several new input representations, which are inspired by the goal of solving both QE and APE with the same model.

2.2 Quality Estimation

Martins et al. (2016) introduced a stacked architecture, using a very large feature set within a structured prediction framework to achieve a large jump in the state of the art for Word-Level QE. Some features are actually the outputs of standalone feedforward and recurrent neural network models, which are then stacked into the final system. Although their approach creates a very good final model, the training and feature extraction steps are quite complicated. An additional disadvantage of this approach is that it requires "jackknifing" the training data for the standalone models that provide features to the stacked model, in order to avoid overfitting in the stacked ensemble. This requires training k versions of each model type, where k is the number of jackknife splits.

Our approach is most similar to Martins et al. (2017), the major differences are: we do not use any internal features from the original MT system, and we do not need to "jackknife" in order to create a stacked ensemble. Using only NMT with attention, we are able to surpass the state-of-the-art in APE and match it in QE.

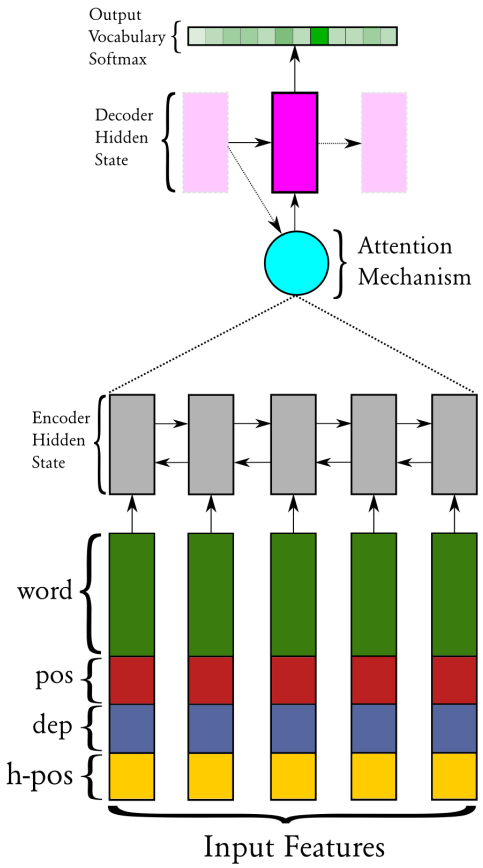

Figure 1: Schematic of the architecture of our factored NMT systems

2.3 Factored Inputs

Alexandrescu and Kirchoff (2006) introduced linguistic factors for neural language models. The core idea is to learn embeddings for linguistic features such as part-of-speech (POS) tags and dependency labels, augmenting the word embeddings of the input with additional features. Recent work has shown that NMT performance can also be improved by concatenating embeddings for additional word-level "factors" to source-word input embeddings (Sennrich and Haddow, 2016). The input representation e_j for each source input x_j with factors F thus becomes Eq. 1:

$$e_j = \Big\|_{k=1}^{|F|} \mathbf{E}_k x_{jk} \quad (1)$$

where $\|$ indicates vector concatenation, $\mathbf{E}_k$ is the embedding matrix of factor k, and x_{jk} is a one hot vector for the k-th input factor.

3 Models

In this section we describe the five model types used for APE-QE, as well as the ensembles of these models which turn out to be the best-performing overall. We design several features to be included as inputs to APE. The operating hypothesis is that that features which haven proven useful for Quality Estimation should also have a positive impact upon APE performance.

Our baseline models are the same models used in Junczys-Dowmunt (2016)[2]. The authors provide trained $SRC \rightarrow PE$ and $MT \rightarrow PE$ models, which correspond to the last four checkpoints from fine-tuning the models on the *500K* training data concatenated with the task internal APE data upsampled 20 times. These models are referred to as **SRC** and **MT**.

3.1 Word Alignments

Previous work has shown that alignment information between source and target is a critical component of current state-of-the-art word level QE systems (Kreutzer et al., 2015; Martins et al., 2016). The sequential inputs for structured prediction, as well as the feedforward and recurrent models in existing work obtain the source-side features for each target word using the word-alignments provided by the WMT task organizers. However, this information is not likely to be available in many real-world usecases for Quality Estimation, and the use of this information also means that the MT system used to produce the hypotheses is not actually a "black box", which is part of the definition of the QE task. Clearly, access to the word-alignment information of an SMT system provides a lot of insight into the underlying model.

Because our models rely upon synthetic training data, and because we wish to view the MT system as a true black-box, we instead use the **SRC** NMT system to obtain these alignments. The attention model for NMT produces a normalized vector of weights at each timestep, where the weights can be viewed as the "alignment probabilities" for each source word (Bahdanau et al., 2014). In order to obtain the input representation shown in table 3, we use the source word with the highest weight from the attention model as an additional factor in the input to another MT-aligned $\rightarrow$ PE system.

[2]These models have been made available by the authors at https://amunmt.github.io/examples/postedit/

WMT 2016 Dev				
Model Input	**BLEU**	**TER ↓**	**F1-Mult**	**Accuracy**
WMT 16 Best	68.94	.215	.493	–
Martins et al (2017)	–	–	**.568**	–
SRC	55.47	.315	.506	.803
MT	66.66	.232	.328	.834
MT-aligned	68.32	.215	.437	.852
SRC+MT	69.17	.211	.477	.857
SRC+MT-factor	69.75	.209	.484	.859
Avg-All Baseline	71.02	.199	.476	.862
Avg-All APE-Tune	**71.22**	**.197**	.510	**.866**
Avg-All QE-Tune	66.92	.228	.554	.857
4-SRC+Avg-All QE-Tune	67.16	.225	.567	.860

WMT 2016 Test				
Model Input	**BLEU**	**TER ↓**	**F1-Mult**	**Accuracy**
WMT Baseline	62.11	.248	.324	–
WMT 16 Best	67.65	.215	.493	–
Martins et al (2017)	67.62	.211	**.575**	–
SRC	55.58	.304	.519	.809
MT	65.85	.234	.347	.837
MT-aligned	67.69	.216	.447	.854
SRC+MT	68.03	.212	.477	.857
SRC+MT-factor	68.28	.211	.473	.857
Avg-All Baseline	**70.05**	.198	.492	.865
Avg-All APE-Tuned	70.04	**.196**	.516	**.868**
Avg-All QE-Tuned	66.93	.219	.573	.864
4-SRC+Avg-All QE-Tune	66.94	.219	**.575**	.865

Table 1: Results for all models and ensembles on WMT 16 development and test datasets

The **MT-aligned** → PE system thus depends upon the **SRC** → PE system to produce the additional alignment factor.

3.2 Inputting Both Source and Target

Following Crego et al. (2016), we train a model which takes the concatenated source and MT as input. The two sequences are separated by a special *BREAK* token. We refer to this system as **SRC+MT**.

3.3 Part-of-Speech and Dependency Labels

Sennrich and Haddow (2016) showed that information such as POS tags, NER labels, and syntactic roles can be included in the input to NMT models, generally improving performance. Inspired by this idea, we select some of the top performing features from Martins et al. (Martins et al., 2016), and include them as input factors to the **SRC+MT-factor** model. The base representation is the concatenated SRC+MT (again with a special *BREAK* token). For each word in the English source and the German hypothesis, we obtain the part-of-speech tag, the dependency relation, and the part-of-speech of the head word, and include these as input factors. For both English and German, we use spaCy[3] to extract these features for all training, development, and test data. The resulting model is illustrated in figure 1.

3.4 Extending Factors to Subword Encoding

Our NMT models use subword encoding (Sennrich et al., 2016), but the additional factors are computed at the word level. Therefore, the factors must also be segmented to match the BPE segmentation. We use the {BILOU}- prefixes common in sequence-labeling tasks such as NER to extend

[3]https://spacy.io/

factor vocabularies and map each word-level factor to the subword segmentation of the source or target text.

Table 3 shows the input representations for each of the model types using an example from the WMT 2016 test data.

3.5 Ensembling NMT Models

We average the parameters of the four best checkpoints of each model type, and create an ensemble of the resulting five models, called **Avg-All Baseline**. We then tune this ensemble for TER (APE) and F1-Mult (QE), using MERT (Och, 2003). The tuned models are called **Avg-All APE-Tuned** and **Avg-All QE-Tuned**, respectively. After observing that source-only models have the best single-model QE performance (see section 5), we created a final F1-Mult tuned ensemble, consisting of the four individual **SRC** models, and the averaged models from each other type (an ensemble of eight models total), called **4-SRC+Avg-All QE-Tune**.

3.6 Tuning

Table 2 shows the final weights for each ensemble type after tuning. In line with the two-model ensemble presented in Martins et al. (2017), tuning models for F1-Mult results in much more weight being allocated to the SRC model, while TER tuning favors models with access to the MT hypothesis.

	APE (TER)	QE (F1-Mult)
SRC	.162	.228
MT	.003	-.183
MT-aligned	.203	.229
SRC+MT	.222	.231
SRC+MT-factor	.410	.129

Table 2: Final weights for each model type after 10 iterations of MERT for tuning objectives TER and F1-Mult.

4 Experiments

All of our models are trained using Nematus (Sennrich et al., 2017). At inference time we use our own decoder, which supports weighted log-linear ensembles of Nematus models[4]. Following Junczys-Dowmunt and Grundkiewicz (2016), we first train each model type on the large (4M) synthetic training data, then fine tune using the 500K dataset, concatenated with the task-internal training data upsampled 20x. Finally, for **SRC+MT** and **SRC+MT-factor** we continued fine-tuning each model for a small number of iterations using the min-risk training implementation available in Nematus (Shen et al., 2016). Table 4 shows the best dev result after each stage of training.

For both APE and QE, we use only the task-specific training data provided for the WMT 2017 APE task, including the extra synthetic training data[5]. However, note that the SpaCy models used to extract features for the factored models are trained with external data – we only use the off-the-shelf models provided by the SpaCy developers.

To convert the output sequence from an APE system into OK, BAD labels for QE, we use the APE hypothesis as a "pseudo-reference", which is then aligned with the original MT hypothesis using TER (Snover et al., 2006).

5 Results

Table 1 shows the results of our experiments using the WMT 16 development and test sets. For each system, we measure performance on BLEU and TER, which are the metrics used in APE task, and also on F1-Mult, which is the primary metric used for the Word Level QE task. Overall tagging accuracy is included as a secondary metric for QE.

All systems with input factors significantly improve APE performance over the baselines. For QE, the trends are less clear, but point to a key difference between optimizing for TER vs. F1_product: F1_product optimization probably lowers the threshold for "changing" a word, as opposed to copying it from the MT hypothesis. This hypothesis is supported by the observation that the source-only APE system outperforms all other single models on the QE metrics. Because the source-only systems cannot resort to copying words from the input, they are forced to make the best guess about the final output, and words which are more likely to be wrong are less likely to be present in the output. If input factors were used with a source-only APE system, the performance on word-level QE could likely be further improved. However, this hypothesis needs more

[4] https://github.com/chrishokamp/constrained_decoding

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

SRC	auto vector masks apply predefined patterns as vector masks to bitmap and vector objects .
MT	automatische Vektor- masken vordefinierten Mustern wie Vektor- masken , Bitmaps und Vektor- objekte anwenden .
MT-aligned	automatische|auto Vektor-|vector masken|masks vordefinierten|apply Mustern|patterns wie|as Vektor-|vector masken|masks ,|to Bitmaps|to und|and Vektor-|vector objekte|objects anwenden|apply .|.
SRC+MT	auto vector masks apply predefined patterns as vector masks to bitmap and vector objects . BREAK automatische Vektor- masken vordefinierten Mustern wie Vektor- masken , Bitmaps und Vektor- objekte anwenden .
SRC+MT Factored	Auto|JJ|amod|NNS vector|NN|compound|NNS masks|NNS|nsubj|VBP apply|VBP|ROOT|VBP predefined|VBN|amod|NNS patterns|NNS|dobj|VBP as|IN|prep|NNS vector|NN|compound|NNS masks|NNS|pobj|IN to|TO|aux|VB bitmap|VB|relcl|NNS and|CC|cc|VB vector|NN|compound|NNS objects|NNS|conj|VB .|.|punct|VBP BREAK|BREAK|BREAK|BREAK Automatische|ADJA|nk|NN Vektor-|B-NN|B-sb|B-VVINF masken|I-NN|I-sb|I-VVINF vordefinierten|ADJA|nk|NN Mustern|NN|pd|NN wie|KOKOM|cd|NN Vektor-|B-NN|B-cj|B-KOKOM masken|I-NN|I-cj|I-KOKOM ,|$,\|punct\|NN Bitmaps\|NN\|cj\|NN und\|KON\|cd\|NN Vektor-\|B-NN\|B-cj\|B-KON objekte\|I-NN\|I-cj\|I-KON anwenden\|VVINF\|ROOT\|VVINF .\|$.|punct|VVINF
PE (Reference)	Automatische Vektormasken wenden vordefinierte Mustern als Vektormasken auf Bitmap- und Vektorobjekte an .
Gold Tags	OK OK BAD OK BAD OK BAD BAD OK OK BAD OK

Table 3: Examples of the input for the five model types used in the APE and QE ensembles. The pipe symbol '|' separates each factor. '-' followed by whitespace indicates segmentation according to the subword encoding.

Model	General	Fine-tune	Min-Risk
MT-aligned	60.31	67.54	–
SRC+MT	59.52	68.68	69.44
SRC+MT-factor	57.59	68.26	69.76

Table 4: Best BLEU score on dev set after each of the training stages. *General* is training with 4M instances, *Fine-tune* is training with 500K + upsampled in-domain data, *Min-Risk* uses the same dataset as *Fine-tune*, but uses a minimum-risk loss with BLEU score as the target metric.

analysis and experimentation to confirm.

6 Conclusion

This work has presented APE-QE, unifying models for APE and word-level QE by leveraging the flexibility of NMT to take advantage of informative features from QE. Models with different input representations are ensembled together and tuned for either APE or QE, achieving state of the art performance in both tasks. The complementary nature of these tasks points to future avenues of exploration, such as joint training using both QE labels and reference translations, as well as the incorporation of other features as input factors.

Acknowledgments

This project has received funding from Science Foundation Ireland in the ADAPT Centre for Digital Content Technology (www.adaptcentre.ie) at Dublin City University funded under the SFI Research Centres Programme (Grant 13/RC/2106) co-funded under the European Regional Development Fund and the European Union Horizon 2020 research and innovation programme under grant agreement 645452 (QT21). Marcin Junczys-Dowmunt provided essential guidance on the tuning implementation for APE and QE ensembles.

References

Andrei Alexandrescu and Katrin Kirchhoff. 2006. Factored neural language models. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*. Association for Computational Linguistics, Stroudsburg, PA, USA, NAACL-Short '06, pages 1–4. http://dl.acm.org/citation.cfm?id=1614049.1614050.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

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*. Association for Computational Linguistics, Lisbon, Portugal, pages 1–46. http://aclweb.org/anthology/W15-3001.

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* .

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *HLT-NAACL*. The Association for Computational Linguistics, pages 866–875.

Spence Green, Jeffrey Heer, and Christopher D. Manning. 2013. The efficacy of human post-editing for language translation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, USA, CHI '13, pages 439–448. https://doi.org/10.1145/2470654.2470718.

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*. Association for Computational Linguistics, Berlin, Germany, pages 751–758. http://www.aclweb.org/anthology/W16-2378.

Julia Kreutzer, Shigehiko Schamoni, and Stefan Riezler. 2015. Quality estimation from scratch (quetch): Deep learning for word-level translation quality estimation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Lisbon, Portugal, pages 316–322. http://aclweb.org/anthology/W15-3037.

Varvara Logacheva, Chris Hokamp, and Lucia Specia. 2016. MARMOT: A toolkit for translation quality estimation at the word level. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation LREC 2016, Portorož, Slovenia, May 23-28, 2016.*. http://www.lrec-conf.org/proceedings/lrec2016/summaries/1054.html.

André F. T. Martins, Ramón Astudillo, Chris Hokamp, and Fabio Kepler. 2016. Unbabel's participation in the wmt16 word-level translation quality estimation shared task. In *Proceedings of the First Conference on Machine Translation*. Association for Computational Linguistics, Berlin, Germany, pages 806–811. http://www.aclweb.org/anthology/W/W16/W16-2387.

André FT Martins, Marcin Junczys-Dowmunt, Fabio N Kepler, Ramón Astudillo, and Chris Hokamp. 2017. Pushing the limits of translation quality estimation .

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proc. of the Annual Meeting on Association for Computational Linguistics*. pages 160–167.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Valencia, Spain, pages 65–68. http://aclweb.org/anthology/E17-3017.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation, WMT 2016, colocated with ACL 2016, August 11-12, Berlin, Germany*. The Association for Computer Linguistics, pages 83–91. http://aclweb.org/anthology/W/W16/W16-2209.pdf.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. http://aclweb.org/anthology/P/P16/P16-1162.pdf.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Michel Simard, Nicola Ueffing, Pierre Isabelle, and Roland Kuhn. 2007. Rule-based translation with statistical phrase-based post-editing. In *Proceedings of the Second Workshop on Statistical Machine Translation*. pages 203–206.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *In Proceedings of Association for Machine Translation in the Americas*. pages 223–231.

Lucia Specia. 2011. Exploiting objective annotations for measuring translation post-editing effort. In *Proceedings of the 15th Conference of the European Association for Machine Translation*. pages 73–80.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*. MIT Press, Cambridge, MA, USA, NIPS'14, pages 3104–3112. http://dl.acm.org/citation.cfm?id=2969033.2969173.

Barret Zoph and Kevin Knight. 2016. Multi-source neural translation. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *NAACL HLT 2016,*

The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016. The Association for Computational Linguistics, pages 30–34. http://aclweb.org/anthology/N/N16/N16-1004.pdf.