Improving Robustness of Neural Machine Translation with Multi-task Learning

Shuyan Zhou, Xiangkai Zeng, Yingqi Zhou
Antonios Anastasopoulos, Graham Neubig

Language Technologies Institute, School of Computer Science
Carnegie Mellon University
{shuyanzh,xiangkaz,yingqiz,aanastas,gneubig}@cs.cmu.edu

Abstract

While neural machine translation (NMT) achieves remarkable performance on clean, in-domain text, performance is known to degrade drastically when facing text which is full of typos, grammatical errors and other varieties of noise. In this work, we propose a multi-task learning algorithm for transformer-based MT systems that is more resilient to this noise. We describe our submission to the WMT 2019 Robustness shared task (Li et al., 2019) based on this method. Our model achieves a BLEU score of 32.8 on the shared task French to English dataset, which is 7.1 BLEU points higher than the baseline vanilla transformer trained with clean text[1].

1 Introduction

Real world data, especially in the realm of social media, often contains noise such as mis-spellings, grammar errors, or lexical variations. Even though humans do not have much difficulty in recognizing and translating noisy or ungrammatical sentences, neural machine translation (NMT; Bahdanau et al. (2015); Vaswani et al. (2017)) systems are known to degrade drastically when confronted with noisy data (Belinkov and Bisk, 2017; Khayrallah and Koehn, 2018; Anastasopoulos et al., 2019). Thus, there is increasing need to build robust NMT systems that are resilient to naturally occurring noise.

In this work, we attempt to enhance the robustness of the NMT system through multi-task learning. Our model is a transformer-based model (Vaswani et al., 2017) augmented with two decoders, with each decoder bound to different learning objectives. It has a cascade architecture (Niehues et al., 2016; Anastasopoulos and Chiang, 2018) where the first decoder reads in the output of the encoder and the second decoder reads in the output of both encoder and the first decoder. The objective of the first decoder, namely the denoising decoder, is to recover from the noisy sentence and generate the corresponding clean sentence. Given both the noisy and clean sentence, the objective of the second decoder, namely the translation decoder, is to correctly translate the sentence to the target language. This framework should be beneficial in two ways: 1) Since the model is trained with noisy text, it should inherently better generalize to noisy text. 2) The translation decoder could potentially take advantage of the recovered clean sentence while maintaining specific varieties of noise (e.g. emoji) by referring to the original noisy sentence. This framework requires triplets of clean and noisy source sentences, along with target translations, so we also follow Vaibhav et al. (2019) and design a back-translation strategy that synthesizes noisy data.

Our proposed model outperforms the baseline vanilla transformer trained with clean text by 4.6 BLEU points on the WMT 2019 Robustness shared task (Li et al., 2019) French to English dataset. The fine-tuning process brings an additional 2.5 points improvement. According to our analysis, however, the improvements can mainly be attributed to introducing noisy data during training rather than the multi-task learning objective.

2 Multi-task Transformer

In this section, we describe in detail the architecture of our proposed multi-task transformer. It is a transformer-based (Vaswani et al., 2017) cascade multi-task framework (Niehues et al., 2016; Anastasopoulos and Chiang, 2018).

2.1 Detailed Architecture

As illustrated in Figure 1, the model consists of one transformer encoder and two transformer de-

[1]The code is available at https://github.com/shuyanzhou/multitask_transformer

Proceedings of the Fourth Conference on Machine Translation (WMT), Volume 2: Shared Task Papers (Day 1) pages 565–571

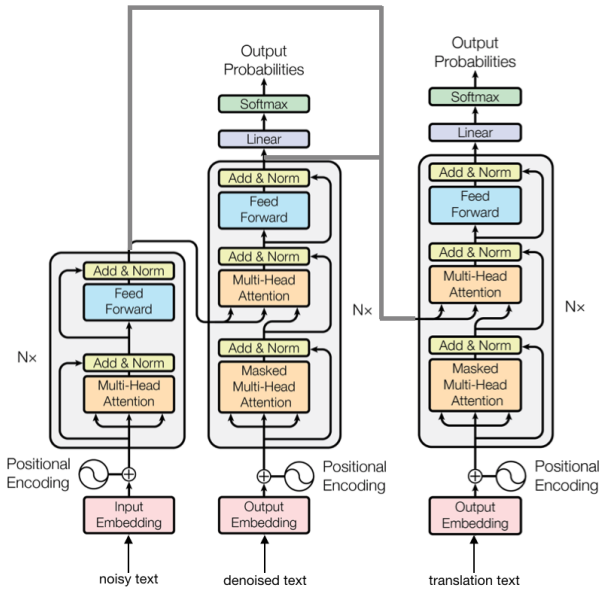

Figure 1: Multitask transformer architecture. Bold grey lines represent parts we add on top of the vanilla transformer.

coders. The dataset consists of triplets: $\mathbf{T} = \{\mathbf{t}_n, \mathbf{t}_c, \mathbf{t}_t\}$ where $\mathbf{t}_n$ is the noisy source sentence, $\mathbf{t}_c$ is the clean source sentence and $\mathbf{t}_t$ is the target translation. Each $\mathbf{t}$ consists of a sequence of words $[w_1, w_2, ..., w_l]$, where l is the length of the corresponding text. By looking up the word and position embedding lookup tables, each $\mathbf{t}$ is converted to a representation matrix $\mathbf{x} = \{e_1, e_2, ..., e_l\}$ and thus result in $\mathbf{X} = \{\mathbf{x}_n, \mathbf{x}_c, \mathbf{x}_t\}$.

The encoder reads in noisy text $\mathbf{x}_n$ and generates the encoded representation $\mathbf{M}_n$. The layers of the first decoder (denoising decoder) first attends to $\mathbf{x}_c$ (self-attention) and then attends to $\mathbf{M}_n$ from the encoder. After N layers, this decoder generates another representation $\mathbf{M}_c$ which represents the clean rather than the noisy source text. Now, the layers of the second decoder (translation decoder) first perform self-attention as usual, and then attend to both $\mathbf{M}_n$ and $\mathbf{M}_c$ simultaneously. After repeating this process N times, the translation decoder generates $\mathbf{M}_t$ which is then passed on to a position-wise feed-forward network followed by a softmax layer. The output of the model is a probability matrix $P \in \mathbb{R}^{l \times V}$, where V is the vocabulary size and l is the length of translated sentence.

As the description above, the denoising decoder is exactly the same as the decoder of the vanilla transformer. The only difference is that for the translation decoder each layer needs to attend to both encoder outputs $\mathbf{M}_n$ and denoising decoder outputs $\mathbf{M}_c$ after self-attention. Therefore, the translation decoder receives two contexts, namely from the encoder attention $\mathbf{A}_n$ and the denoising decoder attention $\mathbf{A}_c$. In our model, we design the final attention context as the linear transformation of the concatenation of these two attention states:

$$\mathbf{A}_t = \mathbf{W}\,[\mathbf{A}_n; \mathbf{A}_c] + \mathbf{b}$$

Where $\mathbf{W} \in \mathbb{R}^{d \times 2d}$ and $\mathbf{b} \in \mathbb{R}^d$.

Following Tu et al. (2017); Anastasopoulos and Chiang (2018), the first objective is to maximize the log likelihood of the clean text $\mathbf{t}_c$ and the second objective is to maximize that of the translated text $\mathbf{t}_t$. The importance of these two objectives are controlled by a hyper-parameter λ:

$$\begin{aligned}\mathcal{L}(\theta) =& \lambda \log P\left(\mathbf{t}_c | \mathbf{t}_n; \theta\right) + \\ & (1-\lambda) \log P\left(\mathbf{t}_t | \mathbf{t}_n, \mathbf{t}_c; \theta\right)\end{aligned} \tag{1}$$

2.2 Two Phase Beam Search

Following Anastasopoulos and Chiang (2018), we use two separate beam search processes to decode the final translation. Let N_{beam} be the size of the beam-search. The process is outlined here for clarity. Given a sentence $\mathbf{t}_n$, the denoising decoder produces a N_{beam} outputs, each consisting of a denoised hypothesis $\hat{\mathbf{t}}_c$, the probability of the hypothesis $P(\hat{\mathbf{t}}_c | \mathbf{x}_n; \theta)$, and corresponding hidden state matrix $\hat{\mathbf{M}}_c$. For each hypothesis from this first decoder, the second decoder also produces N_{beam} tuples, each including a translation hypothesis $\hat{\mathbf{t}}_t$ and its probability $P(\hat{\mathbf{t}}_t | \mathbf{t}_n, \hat{\mathbf{t}_c}; \theta)$. At the end of the second phase, we will have $N_{\text{beam}} \times N_{\text{beam}}$ translation hypotheses. We rank the these hypothesis by their scores defined in Equation 1.

3 Training Triple Generation

As mentioned in Section 2, the desired training data for our multi-task transformer is a collection of triples $\mathbf{T} = \{\mathbf{t}_n, \mathbf{t}_c, \mathbf{t}_t\}$. However, datasets of this kind are very rare; the available amounts of data are less than enough to train such a model with enormous number of parameters. Inspired by Vaibhav et al. (2019), we instead use a back-translation strategy to synthesize these triples. Our proposed strategy is flexible and it could be used as long as we have at least one element of the $\mathbf{T}$ triple.

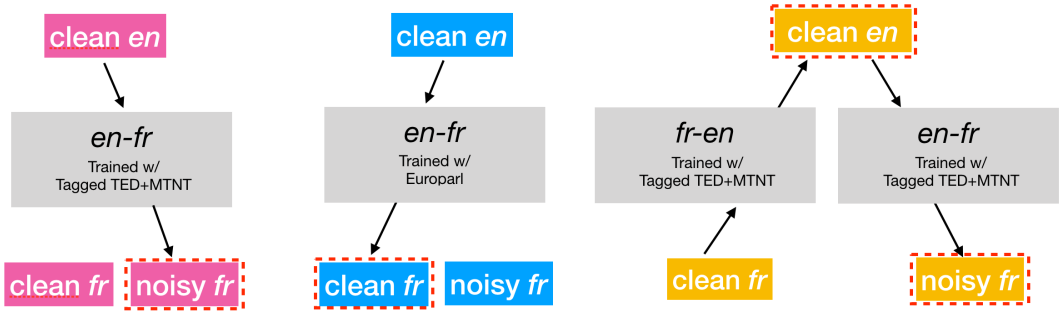

Figure 2: Training data synthesis. Blocks rounded by dash rectangle are synthetic while others are real.

Depending on which part of triple is available, we select the proper NMT model and synthesize the missing ones. In Figure 2, we show 3 ways that we did this in this work. Note that because we focus on the translation from French to English where the French text mostly consists of MTNT-style noise (Michel and Neubig, 2018), we specify the source language as `fr`, the target language as `en` and the noise style as `MTNT`; however, our approach could be used for all other language pairs with different noise distributions.

Clean fr & Clean en: This is the most common parallel corpus that could be obtained from many existing resources. The only missing text is the noisy French text. In this case, we synthesize the noisy text with the help of the NMT model trained with both TED and MTNT training data. During training, we add a tag showing the source of this pair at the beginning of each English sentence (Kobus et al., 2017; Vaibhav et al., 2019). By adding this tag, the model could potentially better distinguish TED data and MTNT data. To generate the noisy French text, we add an MTNT tag at the beginning of each sentence and feed them to this NMT model. Ideally, besides the inherent noise as a result of imperfect translations, the translated French sentences could also possess a similar noise distribution as MTNT.

Noisy fr & Clean en: This kind of parallel text can be found in the MTNT training data. Note that even though the manually translated English sentences contain some level of "noise" (e.g. emoji), we treat them as clean English text. In this scenario, we leverage a pre-trained NMT system provided by fairseq (Ott et al., 2019) to translate English sentences back to French. Considering its good performance over other benchmarks (e.g. WMT newstest datasets) we assume that the translated French sentences are of high quality and thus treat them as clean French text.

Clean fr: To make our back-translation strategy more generalized to settings where the above parallel data is not enough to train the model, we also design a pipeline to utilize monolingual data which is likely to be available most of the time. In this case, we first translate these sentences to English and then translate them back to French. Both NMT models are trained with TED and MTNT data as we describe above. Similarly, in both directions, we add the MTNT tag in the beginning of the sentences. Note that alternatively one could use an off-the-shelf NMT model to generate clean English text.[2]

4 Experiments

In this section, we first describe in detail our data pre-processing scheme, as well as the choice of hyperparameters. Then we compare our system with the baseline model (a vanilla transformer trained on clean French and clean English parallel data). Finally, we carry out a case study by comparing the output of our model with the baseline model.

4.1 Data Pre-processing

Because of time limitations, we did not use all three kinds of training triples. We only used the first two triples introduced in Section 3.

Clean fr & Clean en: The clean data consists of europarl-v7[3] and news-commentary-v10 copora.[4] We filter out sentences whose length is greater

[2] We did not attempt this due to time restrictions.

[3] http://www.statmt.org/europarl/v7/fr-en.tgz

[4] http://www.statmt.org/wmt15/training-parallel-nc-v10.tgz

than 50. We apply a pretrained Byte Pair Encoding (BPE, Gage (1994)) model with 16k subword units to both source and target sentences. The process of synthesizing noisy French sentences is described in the corresponding paragraph of Section 3. We denote this set of triples as $\mathbf{T}_{\text{europarl}}$.

Noisy fr & Clean en: As mentioned in the corresponding paragraph of Section 3, both noisy French and clean English come from MTNT training data and we create clean French through back-translation. This set of triples is denoted as $\mathbf{T}_{\text{mtnt}}$.

4.2 Hyperparameters

We follow the transformer-base setting of Vaswani et al. (2017), using $N = 6$ layers for both encoder and decoder, $h = 8$ heads for self-attention, and d_k, d_v are both set to 64. The hidden size of the model d_{model} is set to 512 and the hidden size of the feed forward network is set to 2048. The smoothing rate ϵ is set to 0.1 and the dropout rate is set to 0.1. For our multi-task transformer specifically, the weight λ in Equation 1 is set to 0.5. The implementation of the model is based on fairseq (Ott et al., 2019)[5].

4.3 Results

The baseline model is the vanilla transformer trained with clean French and clean English. In our experiment, it contains pairs $\mathbf{T}_1 = \{\mathbf{t}_c, \mathbf{t}_t\}$ that are extracted from $\mathbf{X}_{\text{europarl}}$. On the other hand, our model is the multitask transformer trained with $\mathbf{X}_{\text{europarl}}$. The same number of pairs and triples are used during training. We evaluate these two models on two MTNT datasets, one of them comes from the original paper (Michel and Neubig, 2018) while the other one is provided by WMT Robustness shared task (Li et al., 2019). The BLEU score of these two models are shown in the first and the third column of Table 1.

Compared to the vanilla transformer, our proposed multi-task transformer yields 2.5 and 4.6 BLEU points improvement on two MTNT datasets. However, the component that leads to the success of this model is unclear as there are mainly two differences: 1) our proposed model utilizes an auxiliary decoder to recover from the noisy text, it could potentially benefit the translation process with cleaner data 2) our model is further trained on noisy data, presumably overcoming any domain-adaptation issues.

Model	BLEU	
Vanilla Transformer	22.0	25.7
+FT w/ synthetic noise	24.6	27.1
+FT w/ MTNT	**34.1**	**36.0**
Our Model	24.5	30.3
+FT w/ MTNT	31.7	32.8

Table 1: BLEU score of different models. The second column shows the score in MTNT test dataset introduced in Michel and Neubig (2018) and the third column shows the score in the MTNT test dataset provided by WMT Robustness share task (Li et al., 2019).

We investigate this issue by fine-tuning the baseline model with another set of pairs $\mathbf{T}_2 = \{\mathbf{t}_n, \mathbf{t}_t\}$ that are extracted from $\mathbf{T}_{\text{europarl}}$. We load the pre-trained model and continue training for an extra epoch. With this fine-tuning process, the baseline model sees exact the same number of data as our proposed model. The fine-tuning result is shown in the second row of Table 1.

The performance of the fine-tuned baseline system is very close to that of our proposed model on the original MTNT test data and is 3.2 BLEU points lower on the shared task dataset. This result suggest that while the inclusion of synthetic noisy sentences is generalizable among datasets, using the denoising decoder might be beneficial only in specific settings.

Further, to investigate model's potential when in possession of in-domain training data, we fine tune both models with MTNT parallel training data. The data we use here is the same as the MTNT data we use to train auxiliary NMT systems to generate triples (Section 3). During the fine-tuning process, hence, we do not introduce new parallel data. The performance of the fine-tuned systems are shown in the third and the last row of Table 1 respectively.

Even vanilla transformer could not beat the multi-task transformer on both datasets before fine-tuned with in-domain data, it performs significantly better and outperforms our proposed model on both datasets after the fine-tuning process. The results suggest the potential of vanilla transformer in fitting in-domain data. It is notable, of course, that the fine-tuning process leads to a 9.5/8.9 BLEU points improvement for the vanilla transformer and 7.2/1.5 points for our pro-

[5] https://github.com/pytorch/fairseq/tree/master/fairseq

posed model respectively. This again shows the power of domain adaptation for building a robust NMT system.

4.4 Case Study

Table 2 shows example outputs of original MTNT test dataset from different models. The denoised source is the sentence generated by the denoising decoder in our proposed model.

The first example contains special characters '>' and the word 'xQc'. All models fail to correctly copy the special character > and generate a replacement. On the other hand, the word 'xQc' confuses the two baseline models and they fail to correctly copy this word. Our model, however, correctly copies the word and generates a reasonable translation. The denoised sentence seems to not bring benefit and, in fact, it attempts to denoise 'xQc' to 'XVC'. The translation decoder then seems to combine the two versions, copying the word from the source noisy sentence but uppercasing it just like the denoised version.

The second example contains the acronym 'PC' and our model does not produce a correct translation. It is interesting that the translated word 'pellets' is also not the corresponding translation of 'peloton' in the denoised sentence. Somewhat similar to the first example, this suggests that the translation decoder mostly ignores the context from the denoisy decoder. In terms of performance of vanilla transformer, although the baseline model also fails, the fine-tuned model deals with 'PC' correctly and procures a good translation. This indicates that explicitly having attention to both noisy and clean sentences does not always lead to better translation quality.

In the last example, the noise lies in a typo in the phrase corresponding to the phrase ''double negative''. None of the models produces a good translation of this phrase. Similar to the first case, the denoised sentence has a negative effect as it falsely "corrects" ''ngation'' to ''voie'' ("way" in English), which changes the meaning of the word and results in the bad translation 'track'. This demonstrates that all models still need to address issues regarding rare and misspelled words.

The main takeaway from a manual inspection of the outputs, is that the first (denoising) decoder does not really properly deal with noise in the desired way, and the translation decoder generally ignores its output. We suspect that this issue is caused by the data synthesis process which results in low quality triples. Other further improvements could be possibly achieved by constraining the output of the denoising decoder, such that it produces minimal, non-meaning-altering edits. We leave these investigations as future work.

5 Related Work

Here, we discuss how the MT community handles the noise problem. In general, there are mainly two kinds of approaches: the first attempts to denoise text, and the second proposes training with noisy texts.

Denoising text: Sakaguchi et al. (2017) proposes semi-character level recurrent neural network (scRNN) to correct words with scrambling characters. Each word is represented as a vector with elements corresponding to the characters' position. Heigold et al. (2018) investigates the robustness of character-based word embeddings in machine translation against word scrambling and random noise. The experiments show that the noise has a larger influence on character-based models than BPE-based models. To minimize the influence of word structure, Belinkov and Bisk (2017) proposes to represent word as its average character embeddings, which is invariant to these kinds of noise. The proposed method enables the MT system to be more robust to scrambling noise even training the model with clean text. Instead of handling noise at the word level, we try to recover the clean text from the noisy one at the sentence level. Besides noise like word scrambling, the sentence level denoising could potentially better deal with more complex noise like grammatical errors.

Training with noisy data: Li et al. (2017) designs methods to generate noise in the text, mainly focusing on syntactic noise and semantic noise. (Sperber et al., 2017) proposes a noise model based on automatic speech recognizer (ASR) error types, which consists of substitutions, deletions and insertions. Their noise model samples the positions of words that should be altered in the source sentence. Even training with synthetic noise data brings a large improvement in translating noisy data, Belinkov and Bisk (2017) shows that models mainly perform well on the same kind of noise that is introduced at training time, and they mostly fail to generalize to text with other

Source	> Tu veux dire comme xQc?
Target	> Do you mean like xQc?
Baseline	'You want to call it al-Qc?'
Baseline FT	– Do you mean asylum-seekers?
Denoised Source	– Avez-vous lintention de parler de XVC?
Our model	– Do you intend to refer to as XQC?
Source	Si tu joues sur pc, a-t-il t bien adapt?
Target	If you play on PC, has it been well adapted?
Baseline	If you are playing on a pile, has it been adequate?
Baseline FT	If you play on pc, has it been properly adapted?
Denoised Source	Si vous jouez au peloton, a-t-il t bien adapt?
Our model	If you play on pellets, has you been well adapted?
Source	Les franais sont les champions de la double-ngation.
Target	French people are the champions of the double negative.
Baseline	The French are the champions of dual-nation.
Baseline FT	The French are the champions of double-nutrition.
Denoised Source	Les Franais sont les champions de la double voie.
Our model	The French are the champions of the double-track.

Table 2: Comparison of baseline, baseline FT w/ synthetic noise and our model in MTNT fr-en.

kinds of noise. Similar findings were outlined in Anastasopoulos et al. (2019) and Anastasopoulos (2019), which evaluated MT systems on natural and natural-like grammatical noise, specifically on English produced by non-native speakers. Natural noise appears to be richer and more complex compared to synthetic noise, making it challenging to manually design a comprehensive set of noise to approximate real world settings. In our work, we follow (Vaibhav et al., 2019) and synthesize the noisy text through back-translation. There is no need to manually control the distribution of noise.

In terms of multi-task learning for machine translation, Tu et al. (2017) proposes to add a reconstructor on top of the decoder. The auxiliary objective is to reconstruct the source sentence from the hidden layers of the translation decoder. This encourages the decoder to embed complete source information, which helps improve the translation performance. This approach was found to be helpful in low-resource MT scenarios also by Niu et al. (2019). Anastasopoulos and Chiang (2018) proposes a tied multitask learning model architecture to improve the speech translation task. The intuition is that, speech transcription as an intermediate task, should improve the performance of speech translation if the speech translation is based on both the input speech and its transcription.

6 Conclusion

In this work, we propose a multi-task transformer architecture that tries to not only denoisy the noisy source text but also translate it. We design a strategy for synthesizing data triplets for this architecture. Our model could be viewed as a combination of denoising source text and domain adaptation, both of which are popular approaches for designing robust NMT systems. Compared to the baseline vanilla transformer that is trained on clean data only, our proposed model with fine tuning enjoys 7.1 BLEU points improvement on the WMT Robustness shared task French to English dataset. However, this improvement is most likely attributed to the noisy text we add to the training process (hence, due to better domain adaptation), and not due to the denoising multi-task strategy.

Acknowledgements We thank AWS Educate program for donating computational GPU resources used in this work. We also thank Daniel Clothiaux and Junxian He for their insightful comments. This material is based upon work supported in part by the Defense Advanced Research Projects Agency Information Innovation Office (I2O) Low Resource Languages for Emergent Incidents (LORELEI) program under Contract No. HR0011-15-C0114 and the National Science Foundation under grant 1761548. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.

References

Antonios Anastasopoulos. 2019. An analysis of source-side grammatical errors in nmt. In *Proc. BlackboxNLP*. To appear.

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 82–91.

Antonios Anastasopoulos, Alison Lui, Toan Q. Nguyen, and David Chiang. 2019. Neural machine translation of text from non-native speakers. In *Proc. NAACL-HLT*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Georg Heigold, Stalin Varanasi, Günter Neumann, and Josef Genabith. 2018. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, volume 1, pages 68–80.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 74–83.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378.

Xian Li, Paul Michel, Antonios Anastasopoulos, Yonatan Belinkov, Nadir K. Durrani, Orhan Firat, Philipp Koehn, Graham Neubig, Juan M. Pino, and Hassan Sajjad. 2019. Findings of the first shared task on machine translation robustness. In *Proceedings of the Fourth Conference on Machine Translation, Volume 2: Shared Task Papers*, Florence, Italy. Association for Computational Linguistics.

Yitong Li, Trevor Cohn, and Timothy Baldwin. 2017. Robust training under linguistic adversity. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, volume 2, pages 21–27.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836, Osaka, Japan. The COLING 2016 Organizing Committee.

Xing Niu, Weijia Xu, and Marine Carpuat. 2019. Bidirectional differentiable input reconstruction for low-resource neural machine translation. In *Proc. NAACL-HLT*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Matthias Sperber, Jan Niehues, and Alex Waibel. 2017. Toward robust neural machine translation for noisy input sequences. In *International Workshop on Spoken Language Translation (IWSLT), Tokyo, Japan*.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. In *Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL)*, Minneapolis, USA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.