# Improving Robustness of Machine Translation with Synthetic Noise

**Vaibhav*, Sumeet Singh*, Craig Stewart*, Graham Neubig**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
{vvaibhav,sumeets,cas1, gneubig}@cs.cmu.edu

## Abstract

Modern Machine Translation (MT) systems perform remarkably well on clean, in-domain text. However most human generated text, particularly in the realm of social media, is full of typos, slang, dialect, idiolect and other noise which can have a disastrous impact on the accuracy of MT. In this paper we propose methods to enhance the robustness of MT systems by emulating naturally occurring noise in otherwise clean data. Synthesizing noise in this manner we are ultimately able to make a vanilla MT system more resilient to naturally occurring noise, partially mitigating loss in accuracy resulting therefrom [1].

## 1 Introduction

Machine Translation (MT) systems have been shown to exhibit severely degraded performance when required to translate of out-of-domain or noisy data (Luong and Manning, 2015; Sakaguchi et al., 2016; Belinkov and Bisk, 2017). This is particularly pronounced when systems trained on clean, formalized parallel data such as Europarl (Koehn, 2005), are tasked with translation of unedited, human generated text such as is common in domains such as social media, where accurate translation is becoming of widespread relevance (Michel and Neubig, 2018).

Improving the robustness of MT systems to naturally occurring noise presents an important and interesting task. Recent work on MT robustness (Belinkov and Bisk, 2017) has demonstrated the need to build or adapt systems that are resilient to such noise. We approach the problem of adapting to noisy data aiming to answer two primary research questions:

1. Can we artificially synthesize the types of noise common to social media text in otherwise clean data?

2. Are we able to improve the performance of vanilla MT systems on noisy data by leveraging artificially generated noise?

In this work we present two primary methods of synthesizing natural noise, in accordance with the types of noise identified in prior work as naturally occurring in internet and social media based text (Eisenstein, 2013; Michel and Neubig, 2018). Specifically, we introduce a **synthetic noise induction** model which heuristically introduces types of noise unique to social media text and **labeled back translation** (Sennrich et al., 2015a), a data-driven method to emulate target noise.

We present a series of experiments based on the Machine Translation of Noisy Text (MTNT) data set (Michel and Neubig, 2018) through which we demonstrate improved resilience of a vanilla MT system by adaptation using artificially noised data.

## 2 Related Work

Szegedy et al. (2013) demonstrate the fragility of neural networks to noisy input. This fragility has been shown to extend to MT systems (Belinkov and Bisk, 2017; Khayrallah and Koehn, 2018) where both artificial and natural noise are shown to negatively affect performance.

Human generated text on the internet and social media are a particularly rich source of natural noise (Eisenstein, 2013; Baldwin et al., 2015) which causes pronounced problems for MT (Michel and Neubig, 2018).

Robustness to noise in MT can be treated as a domain adaptation problem (Koehn and Knowles, 2017) and several attempts have been made to

* These authors contributed equally
[1]Code available at https://github.com/MysteryVaibhav/robust_mtnt



*Proceedings of NAACL-HLT 2019*, pages 1916–1920
Minneapolis, Minnesota, June 2 - June 7, 2019. 

handle noise from this perspective. Notable approaches (Li et al., 2010; Axelrod et al., 2011) include training on varying amounts of data from the target domain. Luong and Manning (2015) suggest the use of fine-tuning on varying amounts of target domain data, and Barone et al. (2017) note a logarithmic relationship between the amount of data used in fine-tuning and the relative success of MT models.

Other approaches to domain adaptation include weighting of domains in the system objective function (Wang et al., 2017) and specifically curated datasets for adaptation (Blodgett et al., 2017). Kobus et al. (2016) introduce a method of domain tagging to assist neural models in differentiating domains. Whilst the above approaches have shown success in specifically adapting across domains, we contend that adaptation to noise is a nuanced task and treating the problem as a simple domain adaptation task may fail to fully account for the varied types of noise that can occur in internet and social media text.

Experiments that specifically handle noise include text normalization approaches (Baldwin et al., 2015) and (most relevant to our work) the artificial induction of noise in otherwise clean data (Sperber et al., 2017; Belinkov and Bisk, 2017).

## 3 Data

To date, work in the adaptation of MT to natural noise has been restricted by a lack of available parallel data. Michel and Neubig (2018) recently introduced a new data set of noisy social media content and demonstrate the success of fine-tuning which we leverage in the current work. The dataset consists of naturally noisy data from social media sources in both English-French and English-Japanese pairs.

In our experimentation we utilize the subset of the data for English to French which contains data scraped from Reddit[2]. The data set contains training, validation and test data. The training data is used in fine-tuning of our model as outlined below. All results are reported on the MTNT test set for French-English. We additionally use other datasets including Europarl (EP) (Koehn, 2005) and TED talks (TED) (Ye et al., 2018) for training our models as described in §5.

[2] www.reddit.com

| Training Data | # Sentences | Pruned Size |
|---|---|---|
| Europarl (EP) | 2,007,723 | 1,859,898 |
| Ted talk (TED) | 192,304 | 181,582 |
| Noisy Text (MTNT) | 19,161 | 18,112 |

Table 1: Statistics about different datasets used in our experiments. We prune each dataset to retain sentences with length $\leq 50$.

## 4 Baseline Model

Our baseline MT model architecture consists of a bidirectional Long Short-Term Memory (LSTM) network encoder-decoder model with two layers. The hidden and embedding sizes are set to 256 and 512, respectively. We also employ weight-tying (Press and Wolf, 2016) between the embedding layer and projection layer of the decoder.

For expediency and convenience of experimentation we have chosen to deploy a smaller, faster variant of the model used in Michel and Neubig (2018), which allows us to provide comparative results across a variety of settings. Other model parameters reflect the implementation outlined in Michel and Neubig (2018).

In all experimental settings we employ Byte-Pair Encoding (BPE) (Sennrich et al., 2015b) using SentencePiece[3].

## 5 Experimental Approaches

We propose two primary approaches to increasing the resilience of our baseline model to the MTNT data, outlined as follows:

### 5.1 Synthetic Noise Induction (SNI)

For this method, we inject artificial noise in the clean data according to the distribution of types of noise in MTNT specified in Michel and Neubig (2018). For every token we choose to introduce the different types of noise with some probability on both French and English sides in 100k sentences of EP. Specifically, we fix the probabilities of error types as follows: spelling (0.04), profanity (0.007), grammar (0.015) and emoticons (0.002). To simulate spelling errors, we randomly add or drop a character in a given word. For grammar error and profanity, we randomly select and insert a stop word or an expletive and its translation on either side. Similarly for emoticons, we randomly

[3] https://github.com/google/sentencepiece

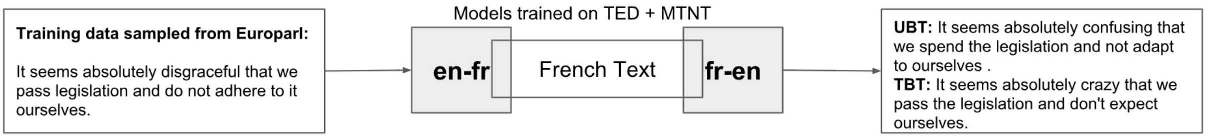


Figure 1: Pipeline for injecting noise through back translation. For demostration purposes we show the process in an English sentence but in experiments, we use French sentences as input.

select an emoticon and insert it on both sides. Algorithm 1 elaborates on this procedure.

**Algorithm 1** Synthetic Noise Induction

**Inputs:** $[(p_1, \eta_1), (p_2, \eta_2) \cdots (p_k, \eta_k)]$ ▷ pairs of noise probabilities and noise functions

**procedure** ADD_NOISE($fr, en$)

$o = 1 - \sum_i p_i$ ▷ probability of keeping original

$D = [o, p_1, p_2, \cdots, p_k]$ ▷ Discrete densities

$j \leftarrow$ SELECT_INDEX(DRAW($D$)) ▷ noise type

**if** $j \neq 0$ **then** ▷ not original

$(fr, en) \leftarrow \eta_j(fr, en)$ ▷ add noise to words

**return** $fr, en$

### 5.2 Noise Generation Through Back-Translation

We further propose two experimental methods to inject noise into clean data using the back-translation technique (Sennrich et al., 2015a).

#### 5.2.1 Un-tagged Back-Translation (UBT)

We first train both our baseline model for fr-en and an en-fr model using TED and MTNT. We subsequently take 100k French sentences from EP and generate a noisy version thereof by passing them sequentially through the trained models as shown in Figure 1. The resulting translation will be inherently noisy as a result of imperfect translation of the intervening MT system.

#### 5.2.2 Tagged Back-Translation (TBT)

The intuition behind this method is to generate noise in clean data whilst leveraging the particular style of the intermediate corpus. Both models are trained using TED and MTNT as in the preceding setting, save that we additionally append a tag in front on every sentence while training to indicate the origin data set of each sentence (Kobus et al., 2016). For generating the noisy version of 100k French sentences from EP, we append MTNT tag in front of the sentences before passing them through the pipeline shown in Figure 1.

| | Training data | BLEU |
|---|---|---|
| *Baselines* | | |
| Baseline | Europarl (EP) | 14.42 |
| + FT w/ | MTNT-train-10k | 22.49 |
| + FT w/ | MTNT-train-20k | 23.74 |
| Baseline FT w/ | TED-100k | 10.92 |
| + FT w/ | MTNT-train-20k | 24.10 |
| *Synthetic Noise Induction* | | |
| Baseline FT w/ | EP-100k-SNI | 13.53 |
| + FT w/ | MTNT-train-10k | 22.67 |
| + FT w/ | MTNT-train-20k | 25.05 |
| *Un-tagged Back Translation* | | |
| Baseline FT w/ | EP-100k-UBT | 18.71 |
| + FT w/ | MTNT-train-10k | 22.75 |
| + FT w/ | MTNT-train-20k | 24.84 |
| *Tagged Back Translation* | | |
| Baseline FT w/ | EP-100k-TBT | 20.49 |
| + FT w/ | MTNT-train-10k | **23.89** |
| + FT w/ | MTNT-train-20k | **25.75** |

Table 2: BLEU scores are reported on MTNT test set. MTNT valid set is used for fine-tuning in all the experiments. + FT denotes fine-tuning of the Baseline model of that particular sub-table, being continued training for 30 epochs or until convergence.

## 6 Results

We present quantitative results of our experiments in Table 2. Of specific note is the apparent correlation between the amount of in-domain training data and the resulting BLEU score. The tagged back-translation technique produces the most pronounced increase in BLEU score of +6.07 points ($14.42 \longrightarrow 20.49$). This represents a particularly significant result given that we do not fine-tune the baseline model on in-domain data, attributing this gain to the quality of the noise generated.

The results for all our proposed experimental methods further imply that out-of-domain clean data can be leveraged to make the existing MT models robust on a noisy dataset. However, sim-

| Systems | Output |
|---|---|
| **REFERENCE** | > And yes, I am an idiot with a telephone in usb-c... F*** that's annoying, I had to invest in new cables when I changed phones. |
| **Baseline (trained on EP)** | And yes, I am an eelot with a phone in the factory ... P***** to do so, I have invested in new words when I have changed telephone. |
| **FT w/ MTNT-train-20k** | > And yes, I am an idiot with a phone in Ub-c. Sh**, it's annoying that, I have to invest in new cable when I changed a phone. |
| **FT w/ EP-100k-TBT** | - And yes, I'm an idiot with a phone in the factory... Puard is annoying that, I have to invest in new cables when I changed phone. |
| **FT w/ EP-100k-TBT + MTNT-train-20k** | > And yes, I am an idiot with a phone in USb-c... Sh** is annoying that, I have to invest in new cables when I changed a phone. |

Table 3: Output comparison of decoded sentences across different models. Profane words are censored.

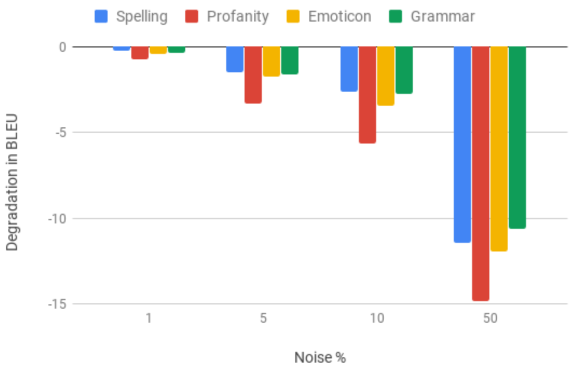


Figure 2: The impact of varying the amount of Synthetic Noise Induction on BLEU.

| Systems | Output |
|---|---|
| **REFERENCE** | Voluntary or not because politicians are *very* friendly with large businesses. |
| **FT w/ EP-100k-TBT** | Whether it's voluntarily, or invoiseally because the fonts are *èsn* friends with the big companies. |
| **FT w/ EP-100k-TBT + MTNT-train-10k** | Whether it's voluntarily, or invokes because the politics are *rès* friends with big companies. |
| **FT w/ EP-100k-TBT + MTNT-train-20k** | Whether it's voluntarily, or invisible because the politics are *very* friends with big companies. |

Table 4: Output comparison of decoded sentences for different amounts of supervision.

ply using clean data is not that beneficial as can be seen from the experiment involving *FT Baseline w/ TED-100k*.

We further present analysis of both methods introduced above. Figure 2 illustrates the relative effect of varying the level of SNI on the BLEU score as evaluated on the newsdiscuss2015[4] dev set, which is a clean dataset. From this we note that the relationship between the amount of noise and the effect on BLEU score appears to be linear. We also note that the most negative effect is obtained by including profanity. Our current approach involves inserting expletives, spelling and grammatical errors at random positions in a given sentence. However we note that our approach might under-represent the nuanced linguistic usage of expletives in natural text, which may result in its above-mentioned effect on accuracy.

Table 3 shows the decoded output produced by different models. We find that the output produced by our best model is reasonably successful at imitating the language and style of the reference. The output of *Baseline + FT w/ EP-100k-TBT* is far superior than that of *Baseline*, which highlights the quality of obtained back translated noisy EP through our tagging method.

We also consider the effect of varying the amount of supervision which is added for fine-tuning the model. From Table 4 we note that the *Baseline + FT w/ EP-100k-TBT* model already produces a reasonable translation for the input sentence. However, if we further fine-tune the model using only 10k MTNT data, we note that the model still struggles with generation of *very*. This error dissipates if we use 20k MTNT data for fine-tuning. These represent small nuances which the model learns to capture with increasing supervision.

To better understand the performance difference between UBT and TBT, we evaluate the noised EP data. Figure 1 shows an example where we can clearly see that the style of translation obtained from TBT is very informal as opposed to the output generated by UBT. Both the outputs are noisy and different from the input but since the TBT method enforces the style of MTNT, the resulting output is perceptibly closer in style to the MTNT equivalent. This difference results in a gain of 0.9 BLEU of TBT over UBT.

## 7 Conclusion

This paper introduced two methods of improving the resilience of vanilla MT systems to noise occurring in internet and social media text: a method of emulating specific types of noise and the use of back-translation to create artificial noise. Both of these methods are shown to increase system accuracy when used in fine-tuning without the need for the training of a new system and for large amounts of naturally noisy parallel data.

[4] http://www.statmt.org/wmt15/test.tgz

## Acknowledgements

The authors would like to thank the AWS Educate program for donating computational GPU resources used in this work.

## References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362. Association for Computational Linguistics.

Timothy Baldwin, Marie-Catherine de Marneffe, Bo Han, Young-Bum Kim, Alan Ritter, and Wei Xu. 2015. Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135. Association for Computational Linguistics.

Antonio Valerio Miceli Barone, Barry Haddow, Ulrich Germann, and Rico Sennrich. 2017. Regularization techniques for fine-tuning in neural machine translation. *CoRR*, abs/1707.09920.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Su Lin Blodgett, Johnny Wei, and Brendan O'Connor. 2017. A dataset and classifier for recognizing social media english. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 56–61. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *HLT-NAACL*.

Huda Khayrallah and Philipp Koehn. 2018. On the impact of various types of noise on neural machine translation. *arXiv preprint arXiv:1805.12282*.

Catherine Kobus, Josep Maria Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *CoRR*, abs/1612.06140.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Conference Proceedings: the tenth Machine Translation Summit*, pages 79–86, Phuket, Thailand. AAMT, AAMT.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *CoRR*, abs/1706.03872.

Mu Li, Yinggong Zhao, Dongdong Zhang, and Ming Zhou. 2010. Adaptive development data selection for log-linear model in statistical machine translation. pages 662–670.

Minh-Thang Luong and Christopher D. Manning. 2015. Neural machine translation systems for spoken language domains.

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*, pages 543–553. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2016. Using the output embedding to improve language models. *CoRR*, abs/1608.05859.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2016. Robsut wrod reocginiton via semi-character recurrent neural network. *CoRR*, abs/1608.02214.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015a. Improving neural machine translation models with monolingual data. *CoRR*, abs/1511.06709.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *CoRR*, abs/1508.07909.

Matthias Sperber, Jan Niehues, and A. Waibel. 2017. Toward robust neural machine translation for noisy input sequences.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. *CoRR*, abs/1312.6199.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488. Association for Computational Linguistics.

Qi Ye, Sachan Devendra, Felix Matthieu, Padmanabhan Sarguna, and Neubig Graham. 2018. When and why are pre-trained word embeddings useful for neural machine translation. In *HLT-NAACL*.