Realistic Noisy Text Generation for Fortified NLP

Danish Pruthi [1] **Sachin Kumar** [1] **Anjalie Field** [1] **Nikolai Vogler** [1]

Abstract

Natural language processing (NLP) systems, especially neural models, are typically trained on carefully controlled clean data. However, in the real world, text is often noisy, which causes NLP systems to perform poorly. One way to alleviate this problem is to train NLP systems on noisy data, but real noisy training data is difficult to obtain and synthetic noise is often very unrealistic. In this project, we propose a VAE model for generating realistic human-like noise that can be used to make NLP systems more robust to noisy real world data. We evaluate the noise generated by our model on how human-like it is and how much it improves downstream NLP tasks. The preliminary improvements achieved by our model and by our baseline models suggest that this is a promising area of research. We further identify several ways to improve our proposed model for future publication.

1. Introduction

Neural models for NLP tasks like machine translation, sentiment analysis, and morphological tagging are typically trained on well-formed data sets (Sperber et al., 2017). As a result, these models are immensely brittle and often perform poorly when confronted with real world *noisy* data (Belinkov & Bisk, 2018; Anastasopoulos et al., 2019; Sperber et al., 2017). Common types of noise in text data include typos, grammatical errors (especially by non-native speakers), and abbreviations or informal content on social media. The poor performance of NLP models in noisy settings has become increasingly problematic with the rapid growth of user-generated content on social media.

One way to alleviate this problem is to explicitly train models on noisy text, thus making them more robust to noisy test data. However, noisy data sets for NLP tasks can be hard to find, especially because the type of noise can differ greatly in different domains. In the absence of existing noisy data, researchers have turned to synthetically introducing noise into training data. While this synthetic noise does improve model performance, it is often very different from real world human-generated noise, and performance improvements essentially disappear when the type of noise in the test data does not match the type of noise in the training data (Belinkov & Bisk, 2018; Karpukhin et al., 2019; Heigold et al., 2017). In this project, we design a model to synthesize human-like noise that can be used during training to improve model robustness.

Prior approaches treat noisy text generation as a translation task, using rule-based heuristics or sequence-to-sequence models to "translate" a clean sentence into a noisy one (Xie et al., 2018; Anastasopoulos et al., 2019). In contrast, we take a generative approach, drawing from prior work on language modeling (Bowman et al., 2016; Guu et al., 2018). Given a clean *prototype* sentence, our proposed model outputs a noisy sentence by drawing a *noise* vector and generating a new sentence conditioned on the noise vector, while attending to the prototype.

Our approach has several advantages in that we can train it on aligned parallel clean/noisy training data or generalize it to non-aligned parallel data; it is more likely to output more diverse types of noise than current approaches; and interpretable hyper-parameters have the potential to control the type and intensity of noise.

We evaluate our model according to two criteria: (1) *How realistic is the generated noise?* We design an annotation task in which we ask annotators to classify whether the noisy text is machine-synthesized or human-generated. Inability of annotators to distinguish would imply that the generated noise is human-like. (2) *How well does our generated noisy data improve downstream NLP tasks?* We use the additional synthesized noisy text to augment training data for machine translation models and test whether the resulting models are robust against naturally occurring noise.

While our evaluations reveal that the noise generated by our model is more human-like than noise generated by rule-based heuristics, one of our baseline models achieves even more human-like noise generation. Similarly, data generated by our model does improve results on a downstream machine translation task, but the success of our baseline models shows the potential for further improvements. We identify areas for improving our proposed model, which we plan to explore in future work.

2. Related Work

Many machine learning models (especially deep learning models) are brittle in the presence of input noise (Biggio et al., 2012; Szegedy et al., 2013). In computer vision, small changes to an input image which are indistinguishable to humans can break image classification systems (Goodfellow et al., 2015). In NLP, input noise degrades the performance of neural models on tasks like machine translation and morphological tagging (Heigold et al., 2017; Anastasopoulos et al., 2019), even though humans show unimpaired comprehension in the presence of such noise (Hahn et al., 2019; Sakaguchi et al., 2017). The surge in user generated content via social media platforms has highlighted the need for NLP models that perform better in this setting.

Improving the robustness of NLP models by training on noisy data has shown promising results. A machine translation model trained on "clean" data, performs much worse when confronted with noisy test data, as opposed to clean test data. However, using rule-based heuristics or random character perturbations to introduce noise in training data reduces the degradation in performance when the test data contains errors by non-native speakers (Anastasopoulos et al., 2019), social media text (Vaibhav et al., 2019), semi-synthetically induced noise (Belinkov & Bisk, 2018), or erroneous output from an automatic speech recognizer (Sperber et al., 2017). Importantly, introducing noise during training is more effective than attempting to reduce noise in test data (Anastasopoulos et al., 2019).

Thus, while training neural models on noisy data is a promising solution, obtaining noisy training sets remains a difficult problem. While small data sets of noisy text have recently been released (Michel & Neubig, 2018; Anastasopoulos et al., 2019), obtaining enough annotated noisy text to train a neural model is a daunting task. Furthermore, unlike for images, where noise can be generated through heuristics like rotations or additive Gaussian noise, it is less obvious how to generate realistic synthetic noise in text. Most current approaches focus on orthographic errors, i.e. making character perturbations such by substituting, inserting, swapping or deleting characters (Heigold et al., 2017). More holistic approaches derive word or phrase-level rules from real-world data, such as changing a verb tense or inserting profanity, and insert these errors into clean training data according to their proportions in the real-word data (Belinkov & Bisk, 2018; Anastasopoulos et al., 2019; Vaibhav et al., 2019; Sperber et al., 2017).

However, all of these methods tend to generate unrealistic noise, subjecting the models to unrealistic scenarios which they might never encounter in the wild. More explicitly, even when trained on noisy data, models perform poorly if the type of noise in the training data does not match the type of noise in the test data (Belinkov & Bisk, 2018; Karpukhin et al., 2019; Sperber et al., 2017). Thus, there is a need for generating noise that is more realistic and human-like.

Vaibhav et al. (2019) suggest back-translation as a method for generating noisy data, which relies on the imperfection of neural machine translation to introduce noise in the text. While they do not evaluate the human-likeness of their generated noisy data, their model trained on the back-translated data does perform the best. Xie et al. (2018) similarly use a back-translation method for generating errors in training data, and also evaluate the human-likeness of the generated noise. However, their focus area is limited to grammatical error correction, and they do not evaluate whether or not this data improves the robustness of NLP models. We use their model as a baseline in our experiments.

The back-translation approaches treat noise generation as a sequence-to-sequence task: given a clean sentence, "translate" it into a noisy sentence. Instead, we draw from work on language modeling and treat noise generation as a language generation task (Bowman et al., 2016). We primarily draw inspiration from Guu et al. (2018), who propose a language generation model that edits a prototype sentence into a new sentence. We specifically aim to edit a prototype sentence into a realistic noisy alternative.

3. Proposed Model

Our overall goal is to learn a generative model that can generate a noisy sentence, given a clean sentence. We describe two different setups. In the first, we assume we have a corpus of clean text and a corpus of noisy text where the sentences are not aligned. In the second, we assume we have sentence-aligned clean and noisy corpora.

3.1. Model for Non-aligned Parallel Training Data

We model the generation of a noisy corpus by assuming each sentence is a modified version of a sentence from a clean corpus. We generate a noisy sentence by:

- Drawing a prototype sentence x' from a distribution over the clean corpus $p(x')$
- Sampling a noise vector z from a noise prior $p(z)$. Then, feeding the edit vector z and the selected prototype x' into a *neural editor* $p(x|x', z)$, which generates a noisy sentence x.

Under this model, the likelihood of a sentence in the noisy corpus is:

$$p(x) = \sum_{x'} p(x|x')p(x') \tag{1}$$

$$p(x|x') = \mathbb{E}_{z\sim p(z)}[p_{\text{noise}}(x|x', z)] \tag{2}$$

A) In conclusion , what I have mentioned above , although global aging issue has become more serious on humans development , if we can balance and understand why we live . B) To sum up , what I have mentioned above , although global aging issue has become more serious on humans development , if we can balance and understand why we live .
A) When these scholars are on field journeys , they also have to relate their observations with their route information , this is when RFID is of great significance . B) When these scholars are on the field journey , they also have to relate their observation with their route information, this is when RFID is of great significance .
A) An ageing population is often viewed negatively by developed countries that are economically oriented . B) Ageing population is often viewed negatively by the developed countries where they are economic driven .
A) A worse thing is that one may not even be aware of that he or she is being tracked . B) One more worse thing is that one may not even be aware of being tracked .

Table 1. Four sample questions from the annotation task. For each pair of sentences, one is human-written and the other one is machine-synthesized. Can you guess which one is human-written? Answers: A, B, B, A.

This model is analogous to the language model proposed by Guu et al. (2018). The main difference is that they train their model by assuming each test sentence x was generated from a prototype sentence x' drawn from the same corpus, whereas we draw prototype sentences from a clean corpus in order to generate noisy sentences in a separate corpus. Thus, while their model learns to generate new clean sentences, ours learns to generate noisy sentences. Nevertheless, we can use the same training procedure derived in Guu et al. (2018), which we briefly summarize here.

The likelihood in Equation 1 cannot be computed exactly because the sum over all possible prototypes (x') is computationally expensive, and the expectation over $p(z)$ has no closed form. We derive two lower bounds to address this problem.

First, rather than considering all possible prototype sentences, for a given noisy sentence x, we define a set of sentences $\mathcal{N}(x)$: the subset of clean sentences that are most similar to x, where we measure similarity through edit distance or Jaccard distance. Then, we can sum over the sentences in $\mathcal{N}(x)$, as the most likely prototypes, rather than summing over all possible clean sentences. Then, we can lower bound Equation 1:

$$\begin{aligned}
\log p(x) &= \log \sum_{x' \in X_{\text{clean}}} p(x|x')p(x') \\
&\geq \log \sum_{x' \in \mathcal{N}(x)} p(x|x')p(x') \\
&= \log \frac{1}{|\mathcal{N}(x)|} \sum_{x' \in \mathcal{N}(x)} p(x|x') + R(x) \\
&\geq \frac{1}{|\mathcal{N}(x)|} \sum_{x' \in \mathcal{N}(x)} \log p(x|x') + R(x)
\end{aligned}$$

Where we take a uniform prior over prototypes

$$p(x') = \frac{1}{|X_{\text{clean}}|}$$

and define

$$R(x) = \log \frac{|\mathcal{N}(x)|}{|X_{\text{clean}}|}.$$

Then, $\sum_{x' \in \mathcal{N}(x)} \log p(x|x')p(x')$ provides a lower bound on $\log p(x)$ up to a constant.

While limiting the set of prototypes reduces the computational complexity of estimating the likelihood, the term $\log p(x|x')p(x')$ is still problematic because it involves an expectation over $p(z)$ that has no closed form (i.e. Equation 2).

We can compute the evidence lower bound (ELBO) over this term by introducing a proposal variational distribution $q(z|x, x')$. Given a clean sentence x' and a noisy sentence x, it generates noise vectors z that are likely to map x' to x. We can then lower bound:

$$\begin{aligned}\log p(x|x') &\geq \mathbb{E}_{z\sim q(z|x',x)}[\log p_{\text{noise}}(x|x',z)] \\ &\quad - \text{KL}(q(z|x',x)||p(z)) \\ &= \text{ELBO}(x,x')\end{aligned}$$

The expectation over $q(z|x,x')$ can be estimated through Monte Carlo approximation. $q(z|x,x')$ and $p_{\text{noise}}(x|x',z)$ together form a variation autoencoder (VAE) (Kingma & Welling, 2013). The final objective function, $\sum_{x'\in\mathcal{N}(x)} \text{ELBO}(x,x')$ can be optimized through stochastic gradient ascent.

3.2. Model Architecture

We use the same architecture as Guu et al. (2018). For $p_{\text{noise}}(x,x',z)$, we use an encoder-decoder model with attention, where x' is the input and x is the output, and the conditional vector z is concatenated to the input of the decoder at each time step.

For $p(z)$, we sample a scalar length $z_{\text{norm}} \sim \text{Unif}(0,10)$ and a direction z_{dir} from the uniform distribution on the unit sphere and set $z = z_{\text{norm}} \cdot z_{\text{dir}}$.

For the proposal distribution $q(z|x',x)$, we derive a vector that reflects the difference between x' and x and add perturbations to generate samples. More specifically, we assume x' can be transformed into x by deleting a set of words D and inserting a set of words I. We define:

$$f(x,x') = \sum_{w\in I} \Phi(w) \oplus \sum_{w\in D} \Phi(w)$$

where $\Phi(w)$ denotes a word embedding for w (initialized with pretrained embeddings). Then we use von-Miser Fisher (vMF) noise to perturb the direction of f and uniform noise to perturb the magnitude. More explicitly, $z = z_{\text{dir}} \cdot z_{\text{norm}}$, where z_{dir} is sampled from a vMF distribution with mean direction f and concentration parameter κ, and z_{norm} is sampled from a uniform distribution, parametized by $||f||$ and $||f||+\epsilon$. Thus the parameters for q are the word embeddings Φ and the hyperparameters ϵ and κ.

We refer to Guu et al. (2018) for the gradient computations, which fully specify the training procedure.

3.3. Modifications for Aligned Parallel Training Data

In the second setup, we assume that we have a clean corpus and a noisy corpus where each noisy sentence is aligned to a clean sentence. In our model, because we can assume each noisy sentence was generated from its parallel-aligned clean counterpart, we can skip the summation in Equation 1, and the likelihood becomes:

$$p(x) = \mathbb{E}_{z\sim p(z)}[p_{\text{noise}}(x|x',z)] \tag{3}$$

3.4. Model Justification

Our model has the potential to improve over prior work for several reasons. First, Xie et al. (2018) show that sequence-to-sequence models can successfully generate human-like noise. This type of model, where how to modify a clean sentence into a noisy sentence is implicitly learned, rather than determined through pre-set rules as in (Anastasopoulos et al., 2019; Vaibhav et al., 2019), has the potential to generalize to different types of noise and different languages. However, their model relies on parallel-aligned data, which is not available for most types of noise, i.e. social media. The first variant of our model does not rely on parallel-aligned data, and thus can be used in a wide variety of settings.

In the case that our non-aligned variant performs poorly, our aligned variant still has the potential to improve over existing baselines. We believe that implicitly learning edit variants as latent vectors z better reflects the relationship between clean and noisy texts than sequence-to-sequence translation. Additionally, Guu et al. (2018) show that in the context of language modeling, the latent vectors z encode meaningful semantic information. In our model, because we train on clean-to-noisy text, we expect the z vectors to meaningfully encode information about the type of noise generated. This representation would allow us a significant amount of control over the type of noise generated. We can apply a specific noise variant to a clean sentence s by taking a sentence pair x' and x with the desired noise type, learning a latent vector z for the $x' \rightarrow x$ transformation, and using z to transform s. Thus, our model can mimic rule-based approaches, but we learn rules rather than pre-defining them. In this way, it combines the controllability of rule-based approaches with the flexibility of the sequence-to-sequence approach.

4. Experiments

The full pipeline involves first training the noise-generation model on a data set containing parallel clean and noisy data (either aligned or not aligned), and then using the trained model to introduce noise into a separate data set.

4.1. Experimental Setup

We evaluate our model's ability to generate noise through two criteria:

How realistic is the generated noise? We design an annotation task comparable to the annotations conducted in Xie et al. (2018). For data sets with aligned noisy/clean pairs, annotators will be presented with two noisy sentences —

one generated by our model and the other from the set of actual errors made by non-native speakers. Annotators will be asked to distinguish which sentence was generated by a machine. A low accuracy on these tasks would indicate that our model generates realistic human-like noise.

How well does our generated noisy data improve downstream NLP tasks? We evaluate our model on machine translation of noisy text. Specifically, we train a machine translation system on a clean data set and evaluate performance on noisy datasets. Then, we use our model to generate noise in the training set, retrain the machine translation model on the augmented training data, and evaluate the change in performance on the noisy test data.

4.2. Aligned Data (English-Spanish Translation)

For parallel-aligned training data, we need a data set containing clean and noisy versions of the same sentences. We use the Lang-8 corpus (Tajiri et al., 2012), which contains 1 million aligned pairs, matching a text containing errors to a corrected version.[1]

After training our noise generation model on this data set, we use it to generate noisy training data for an English-Spanish translation task. The translation task uses the JFLEG-es corpus as the noisy test data. This data set consists of 747 dev and 754 test sentences in English written by non-native speakers, drawn from the JHU FLuency-Extended GUG corpus (JFLEG) and then translated into Spanish by professional translators. We refer to this test set as JFLEG-ES (Anastasopoulos et al., 2019). In both the Lang-8 and the JFLEG-ES corpora, the noise in this data consists of errors made by non-native English speakers. Thus, we match the type of noisy data used to train our noise generator with the type of noise we expect to encounter in the translation task.

In order to train an English-Spanish translation model, we use the Europarl (Koehn, 2005) corpus as clean training data, which contains 2M training instances. Specifically, we use newstest2013 and newstest2014 as clean dev and test sets respectively. Thus our full pipeline involves training the noise generation model on the Lang 8 corpus, using the model to generate noise in the Europarl training data, augmenting the clean data with the generated noisy data for training the English-Spanish translation model, and evaluating the translation model on the JFLEG-ES corpora.

4.3. Non-aligned Data (English-French Translation)

As in the aligned case, for non-aligned training data we need a corpus of clean and noisy data, where the type of noise matches the type of noise we expect to see in our downstream evaluation task: in our case, we choose a social media setting. However, we do not mandate that the corpus be sentence-aligned. In order to meet these requirements, we gather a new corpus, where the clean data consists of newspaper articles and the noisy data consists of social media posts linking the articles. More specifically, we scrape posts and comments from Reddit that link newspaper articles (from subreddits like r/news, r/worldnews, r/politics which have posts from the same domain as the MT datasets we use). Following Guu et al. (2018), we use metrics like Jaccard similarity and Levenshtein edit distance to roughly align each noisy sentence (from Reddit) to a set of clean prototype sentences (from newspaper articles).

We originally intended to train our noise generation model on this new data set and evaluation on a English-French translation task, using the MTNT data set (Michel & Neubig, 2018). This data set contains 1,020 test sentences and 852 validation sentences from English social media translated into French. This data set also contains 36,058 training samples, which is too small to train an effective neural translation model from scratch but can be used for fine-tuning (Vaibhav et al., 2019).

As we discuss in 5, we judged that our Reddit/newspaper data set captured noise too poorly to sufficiently train our model, and we ultimately did not conduct this evaluation.

4.4. Experimental Setup

All the data for MT experiments is tokenized and truecased using Moses[2] and split into subwords using Byte Pair Encoding (BPE) with 32,000 operations (Sennrich et al., 2015). We filter the training set to contain a maximum of 80 words per sentence. We use a fully convolutional encoder and decoder model (Gehring et al., 2017) to train all our MT systems implemented in Pytorch[3] with the recommended settings for en-de provided by the authors.

We use two noising models from prior work as baselines for comparison. The first, proposed by Anastasopoulos et al. (2019), involves learning distributions of error types from a noisy data set, and then using a rule-based approach to proportionally introduce noise into a separate data set. We refer to this approach as "synth-noise". The second, proposed by Xie et al. (2018), uses a back-translation approach, where a sequence-to-sequence model is trained to "translate" a clean sentence into a noisy one using parallel-aligned training data, and beam search noising is used to encourage diversity in outputs. We refer to this approach as "back-translate-noise".

As a sanity check on our implementations, we also evaluate our models on a grammatical error correction task, following Xie et al. (2018).[4] We use the noisy data generated

[1] http://lang-8.com/

[2] https://github.com/moses-smt/mosesdecoder

[3] https://github.com/pytorch/fairseq/

[4] We find it difficult to reliably reproduce results from the origi-

by our baseline models as well as the clean input text they were generated from as auxiliary training data to train a model for grammatical error correction. For this task, we also use a fully convolutional encoder-decoder with network size and hyperparameters recommended by Chollampatt & Ng (2018)[5]. Contrary to Xie et al. (2018), we encode and decode with BPE subword units instead of characters since they have been shown to perform better at sequence-to-sequence tasks (Sennrich et al., 2015). Similar to Xie et al. (2018), we use the same network structure and pre-processing procedure to generate noisy examples for our "back-translate-noise" setting. We report results in Table 2. As expected, augmenting the training data with our generated noise improves performance on this task, which verifies our implementations of the baseline models. We do not use this task as an evaluation metric for our proposed model because our goal is to evaluate how well noise generated by our systems improve unrelated NLP tasks, rather than a task that is specifically focused on noise correction.

Noising Method	# Train Sent.	$F_{0.5}$
None	1.3M	30.52
Synth (Anastasopoulos et al., 2019)	2.3M	**33.48**
Back-translate (Xie et al., 2018)	2.3M	31.81

Table 2. GEC Results on CoNLL 2014 Dev and Test set. We explore the benefit of augmenting synthetic noisy examples from two different noising models.

Noising Method	Annotator Accuracy
Synth (Anastasopoulos et al., 2019)	75%
Back-translate (Xie et al., 2018)	41%
Ours	64%

Table 3. Evaluation of human-likeness of noise over 100 generated samples. $< 50\%$ accuracy suggests that the model generated human-like noise

5. Results and Discussion

5.1. Human-likeness of Noise

In a human study on 100 sentence pairs, we compare the noisy text written by non-native speakers with machine synthesized noise. We show a few samples from the annotation task in Table 1; the noisy examples in this table were generated from the back-translate model. We additionally show sample outputs from our proposed model and the Synth model in Table 4. In the examples from the synth model, we can see the stilted effects of the rule-based translation: dropping random letters like "bad → bd" or changing the forms of verbs like "is → am" results in unrealistic errors. In contrast, our proposed model has greater fluency.

This observation is reflected in the results from our annotation task, show in Table 3. Annotators were able to distinguish outputs from the synthetic model from real sentences with high accuracy (75%). In contrast, outputs from the proposed model were harder to distinguish from real sentences (64% accuracy). However, annotator accuracy was lowest for the back-translate model (41%), in which human annotators performed worse than random guessing.

In examining the output of the proposed model, we identified several areas for improvement. First, out of the 100 randomly sampled examples, in 17 cases our proposed model generated the exact same text as the human-written reference sentences. In these cases, the annotator randomly guesses which sentences was written by a human, since both sentences are identical. When we remove these 17 matching examples from consideration, the annotator accuracy remains approximately the same 63.9%; however, the frequency with which our model imitates human text exactly suggests it has the potential to produce noise that is very human-like. In contrast, the back-translate model produced the same output as the human text only 3/100 times. However, the proposed model is more likely to make egregious errors than the back-translate model, such as dropping large parts of the sentence. We also observed evidence of posterior mode collapse, which is a common problem in generative models. Mode collapse refers to a generative model ignoring the latent variables and relying too heavily on the prior. In our model, this is evidenced by the model producing output sentences that are unrelated to the inputs sentences. We show examples of this in Table 5. Based on these observations, we hypothesize that the proposed model has the potential to outperform the back-translate model, but that it requires additional modifications and fine tuning to prevent mode collapse and reduce the number of erroneous outputs. Recent research on ways to prevent model collapse in VAEs offers a starting point for improving our model (He et al., 2019).

Another problem we encountered with the proposed model is the frequency of UNK tokens in the output. In training the model, we restrict the vocabulary to reduce computational complexity and replace out of vocabulary words with UNK tokens. These UNK tokens are then still present in the output of the model. We use a simple heuristic to replace UNK tokens with the original out of vocabulary words from the input sentence, but this method is imperfect and could impact results.

Finally, we also note that we had different annotators complete different parts of the annotation task, and some annotators may have been better at identifying noise than others, which may have biased these results.

nal paper as the authors do not disclose crucial noising parameters that greatly affect the output quality.

[5] https://github.com/nusnlp/mlconvgec2018

Proposed Model
One may then question the result.
We will examine as part of the review.
There is no doubt that your report will be voted for the European parliament.
Synth Model
Another thing : these new countries include the Cyprus.
Parliament am being slated for this from outside.
No deal is better than a bd deal.

Table 4. Sample "noised" sentences, outputted by our proposed model and by the synthetic baseline

Input	**Output**
This poet is the symbol of Palestinian patriotism	in addition, people over 100 people were wounded.
Europe has to decide what kind of solidarity is necessary.	it is empty words and no action.
I was among those who expressed their concern.	therefore i did not for the resolution.

Table 5. Sample output sentences generated by the proposed model. These examples show evidence of mode collapse, where the model ignores the input sentence

Training Set	**Test Set**	
	Clean	**Noisy**
Clean	28.50	22.3
Clean+Back-translate	28.9	22.9
Clean+Synth	29.3	22.4
Clean+Ours	29.4	22.5

Table 6. BLEU scores obtained on the newstest2014 and JFLEG-es sets when trained with clean data (2M) and clean data augmented with noisy data(3.5M) generated using the two baseline models and our proposed model

5.2. Aligned Data (English-Spanish Translation)

Table 6 reports BLEU scores over the JFLEG-es dataset. For comparison, we also show BLEU scores over a clean test set (newstest2014). BLEU score was computed using SacreBLEU script[6]. First, we show the decline in BLEU score (from 28.50 to 22.3) when the test set is clean as compared to when the test set contains noise, which exemplifies the brittleness of NLP systems when confronted with noise data and motivates our work. As described in Section 4, we then augment the clean (Europarl) training data with noise generated by the three noise generation models and train the machine translation system on the augmented data.

Augmenting the training data with the output of the back-translate model gives an improvement of 0.6 BLEU points (22.3 to 22.9). We note that while we refer to the back-translate model as a baseline, this model has not previously been used for this task, as it was initially designed exclusively for grammatical error correction. Surprisingly, augmenting training data with outputs from the back-translate model also improves BLEU by 0.4 points even when the test data is not noisy (from 28.5 to 28.9).

The synthetic model improves BLEU trivially (0.1 points) over the noisy test data. This result does not match the numbers reported in Anastasopoulos et al. (2019), who observed an 0.6 improvement in BLEU. Their translation system architecture differs from ours and they also introduce 5x more noise examples into the corpus, which could account for the difference in result. However, our back-translate system does match their reported results.

Our proposed model improves performance more than the synthetic baseline but does not perform as well as the back-translate baseline. However, as discussed in Section 5.1, we identified mode collapse and UNK replacement heuristics as potential problems with our model. We suspect that if we address these issues, we can improve performance beyond the back-translate baseline.

5.3. Generation of Social Media Noisy Data

We attempted to collect a corpus containing noisy sentences from social media roughly aligned to clean prototype sentences from newspaper articles. We intended to use this data set to train our noise generation models, and then evaluate how well the generated noise improves machine translation of social media by using the MTNT data set (Michel & Neubig, 2018).

We scraped posts from r/news, r/Politics, r/InTheNews, r/todayilearned, and we further scraped the text of all newspaper articles linked in the collected posts. Then, for a sentence from a social media post, we used Jaccard similarity (the same metric used by (Guu et al., 2018) to identify

[6]https://github.com/mjpost/sacreBLEU

Clean	Noisy
and they may have bristled at what they heard.	* * and they may have bristled at what they heard
but the other side of this is the benefit to fox news.	the other side of this is the benefit to fox news.
I'm not even a big believer in democracy.	not a believer in democracy?
let me repeat: theres nothing wrong with being successful.	theres nothing wrong with being successful.

Table 7. Sample noisy sentences from r/Politics matched with clean sentences from newspaper articles. The alignment primarily captures partial quotes

prototype sentences) in order to identify related sentences from newspaper articles that were linked in the same thread as the social media post.

We encountered several problems with this approach. First, the number of matched sentences obtained is very low. For instance, 42,199 comments from r/Politics yields 395 matched sentences. Second, we judged the quality of the matches to be very poor. We show examples in Table 7. Many of them consist of partial quotes, where the social media user quoted part of sentences from a newspaper article. Thus, even though we do filter out exactly matching sentences, we do not catch these partial quotes. We experimented with different thresholds for Jaccard similarity as well as alternate similarity metrics (Levenstein edit distance), but we were unable to generate a data set that we judged was representative of social media style noise.

We intend to explore alternative potential sources of social media style noise, for instance data sets from Twitter where the same user reposts the same URL multiple times with different text (Tan et al., 2014).

6. Conclusions and Future Work

We propose a new model for generating human-like noisy text, which can be used to train NLP systems and increase their robustness to noisy test data. We evaluate our proposed model as well as several baselines on a machine translation task. The preliminary success of our proposed model as well as the back-translate baseline shows the potential benefits of our work.

Furthermore, we have identified a few specific areas for improving our model. First, we believe posterior mode collapse may be one of the main problems with the model, and we plan to investigate ways to mitigate this issue (He et al., 2019). Additionally, we believe the main advantage of our model is that we can generalize it to non-aligned data sets, which we were unable to test in this project due to the difficulty of data collection. However, as discussed in Section 5.3, we intend to explore additional ideas for data collection in order to test our model on types of noise other than errors made by non-native speakers.

Finally, the other main advantage of a VAE model is that it learns latent z noise vectors, which we expect to encode meaningful information about the type of noise generated. In theory, these vectors can allow us a significant amount of control over the type of noise generated. While we were not able to explore this idea because of the poor performance of our model, we hope that after improving model performance, we can exploit these latent vectors to interpret and control the process of noise generation.

References

Anastasopoulos, A., Lui, A., Nguyen, T. Q., and Chiang, D. Neural machine translation of text from non-native speakers. In *Proc. NAACL HLT*, 2019. to appear.

Belinkov, Y. and Bisk, Y. Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*, 2018.

Biggio, B., Nelson, B., and Laskov, P. Poisoning attacks against support vector machines. In *Proc. ICML*, 2012.

Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., and Bengio, S. Generating sentences from a continuous space. *CoNLL 2016*, pp. 10, 2016.

Chollampatt, S. and Ng, H. T. A multilayer convolutional encoder-decoder neural network for grammatical error correction. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. Convolutional sequence to sequence learning. *CoRR*, abs/1705.03122, 2017.

Goodfellow, I., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL http://arxiv.org/abs/1412.6572.

Guu, K., Hashimoto, T. B., Oren, Y., and Liang, P. Generating sentences by editing prototypes. *Transactions of the Association of Computational Linguistics*, 6:437–450, 2018.

Hahn, M., Keller, F., Bisk, Y., and Belinkov, Y. Character-based surprisal as a model of human reading in the presence of errors. *arXiv preprint arXiv:1902.00595*, 2019.

He, J., Spokoyny, D., Neubig, G., and Berg-Kirkpatrick, T. Lagging inference networks and posterior collapse in variational autoencoders. In *ICLR*, 2019.

Heigold, G., Neumann, G., and van Genabith, J. How robust are character-based word embeddings in tagging and MT against wrod scramlbing or randdm nouse? *CoRR*, abs/1704.04441, 2017. URL http://arxiv.org/abs/1704.04441.

Karpukhin, V., Levy, O., Eisenstein, J., and Ghazvininejad, M. Training on synthetic noise improves robustness to natural noise in machine translation. *arXiv preprint arXiv:1902.01509*, 2019.

Kingma, D. P. and Welling, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Koehn, P. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pp. 79–86, 2005.

Michel, P. and Neubig, G. Mtnt: A testbed for machine translation of noisy text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 543–553, 2018.

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

Sennrich, R., Haddow, B., and Birch, A. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*, 2015.

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

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. *CoRR*, 2013. URL http://arxiv.org/abs/1312.6199.

Tajiri, T., Komachi, M., and Matsumoto, Y. Tense and aspect error correction for esl learners using global context. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pp. 198–202. Association for Computational Linguistics, 2012.

Tan, C., Lee, L., and Pang, B. The effect of wording on message propagation: Topic- and author-controlled natural experiments on twitter. In *Proceedings of ACL*, 2014.

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

Xie, Z., Genthial, G., Xie, S., Ng, A., and Jurafsky, D. Noising and denoising natural language: Diverse backtranslation for grammar correction. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, volume 1, pp. 619–628, 2018.