Training on Synthetic Noise Improves Robustness to Natural Noise in Machine Translation

Vladimir Karpukhin **Omer Levy** **Jacob Eisenstein**[*] **Marjan Ghazvininejad**

Facebook Artificial Intelligence Research
Seattle, WA

Abstract

Contemporary machine translation systems achieve greater coverage by applying subword models such as BPE and character-level CNNs, but these methods are highly sensitive to orthographical variations such as spelling mistakes. We show how training on a mild amount of random synthetic noise can dramatically improve robustness to these variations, without diminishing performance on clean text. We focus on translation performance on natural typos, and show that robustness to such noise can be achieved using a balanced diet of simple synthetic noises at training time, without access to the natural noise data or distribution.

1 Introduction

Machine translation systems are generally trained on clean data, without spelling errors. Yet many translation scenarios require robustness to such errors: for example, social media text in which there is little emphasis on standard spelling (Michel and Neubig, 2018), and interactive settings in which users must enter text on a mobile device. Systems trained on clean data generally perform poorly when faced with such errors at test time (Heigold et al., 2017; Belinkov and Bisk, 2018).

One potential solution is to introduce noise at training time, similar in spirit to the use of adversarial examples (Goodfellow et al., 2014; Ebrahimi et al., 2018). So far, using synthetic noise at training time has been found to improve performance only on test data with exactly the same kind of synthetic noise, while at the same time impairing performance on *clean* test data (Heigold et al., 2017; Belinkov and Bisk, 2018). We desire methods that perform well on both clean text and naturally-occurring noise, but this is beyond the current state of the art.

Drawing inspiration from dropout and noise-based regularization methods, we explore the space of random noising methods at training time, and evaluate performance on both clean text and text corrupted by “natural noise” found in real spelling errors. We find that by feeding our translation models a balanced diet of several types of synthetic noise at training time (random character deletions, insertions, substitutions, and swaps), it is possible to obtain substantial improvements on such naturally noisy data, with minimal impact on the performance on clean data, and without accessing the test noise data or even its distribution.

Our method substantially improves the robustness of a transformer-based machine translation model with CNN character encoders to spelling errors across multiple input languages (German, French, and Czech). Of the different noise types we use at training, we find that random character deletions are particularly useful, followed by character insertions. However, noisy training does not improve translations of social media text, as indicated by performance on the MTNT dataset of Reddit posts (Michel and Neubig, 2018). This finding aligns with previous work arguing that the distinctive feature of social media text is not noise or orthographical errors, but rather, variation in writing style and vocabulary (Eisenstein, 2013).

2 Noise Models

We focus on *orthographical noise*; character-level noise that affects the spelling of individual terms. Orthographical noise is problematic for machine translation systems that operate on token-level embeddings because noised terms are usually out-of-vocabulary, even when divided into subwords using techniques such as byte pair encoding (BPE; Sennrich et al., 2015). Interestingly, orthographical noise can also pose problems for *character-*

[*] Jacob Eisenstein is now at Google Research.

Proceedings of the 2019 EMNLP Workshop W-NUT: The 5th Workshop on Noisy User-generated Text, pages 42–47
Hong Kong, Nov 4, 2019. ©2019 Association for Computational Linguistics

Deletion	A character is deleted.	whale → whle
Insertion	A character is inserted into a random position.	whale → wxhale
Substitution	A character is replaced with a random character.	whale → whalz
Swap	Two adjacent characters change position.	whale → wahle

Table 1: The synthetic noise types applied during training. Noise is applied on a random character, selected from a uniform distribution. The right column illustrates the application of each noise type on the word "whale."

level encoding models, which are based on models such as convolutional neural networks (CNNs; Kim et al., 2016). These models learn to match filters against specific character n-grams, so when n-grams are disrupted by orthographical noise, the resulting encoding may radically differ from the encoding of a "clean" version of the same text. Belinkov and Bisk (2018) report significant degradations in performance after applying noise to only a small fraction of input tokens.

Synthetic Noise Table 1 describes the four types of synthetic noise we used during training. Substitutions and swaps were experimented with extensively in previous work (Heigold et al., 2017; Belinkov and Bisk, 2018), but deletion and insertion were not. Deletion and insertion pose a different challenge to character encoders, because they alter the distances between character sequences in the word, as well as the overall word length.

During training, we noised each token by sampling from a multinomial distribution of 60% clean (no noise) and 10% probability for each of the four noise types. The noise was added dynamically, allowing for different mutations of the same example over different epochs.

Natural Noise We evaluate our models on *natural* noise from edit histories of Wikipedia (for French and German; Max and Wisniewski, 2010; Zesch, 2012) and manually-corrected essays (for Czech; Šebesta et al., 2017). These authors have obtained a set of likely spelling error pairs, each involving a clean spelling and a candidate error. We used that set to replace correct words with their misspelled versions for each evaluation sample text in the source language. When there are multiple error forms for a single word, an error is selected randomly. Not all words have errors, and so even with maximal noise, only 20-50% of the tokens are noised.

Natural noise is more representative of what might actually be encountered by a deployed machine translation system, so we reserve it for test data. While it is possible, in theory, to use natural noise for training, it is not always realistic. Significant engineering effort is required to obtain such noise examples, making it difficult to build naturally-noised training sets for any source language. Furthermore, orthography varies across demographics and periods, so it is unrealistic to anticipate the exact distribution of noise at test time.

3 Experiment

Data Following Belinkov and Bisk (2018), we evaluated our method on the IWSLT 2016 machine translation benchmark (Cettolo et al., 2016). We translated from three source languages (German, French, Czech) to English, each with a training set of approximately 200K sentence pairs. Synthetic noise was added only to the training data, and natural noise was added only to the test data; the validation data remained untouched.

Model We used a transformer-based translation model (Vaswani et al., 2017) with a CNN-based character encoder (Kim et al., 2016).

Hyperparameters We followed the base configuration of the transformer (Vaswani et al., 2017) with 6 encoder and decoder layers of 512/2048 model/hidden dimensions and 8 attention heads. Character embeddings had 256 dimensions and the character CNN followed the specifications of Kim et al. (2016). We optimized the model with Adam and used the inverse square-root learning rate schedule typically used for transformers, but with a peek learning rate of 0.001. Each batch contained a maximum of 8,000 tokens. We used a dropout rate of 0.2. We generated the translations with beam search (5 beams), and computed BLEU scores to measure test set performance.

Results Table 2 shows the model's performance on data with varying amounts of natural errors. As observed in prior art (Heigold et al., 2017; Belinkov and Bisk, 2018), when there are significant amounts of natural noise, the model's performance drops significantly. However, training on our synthetic noise cocktail greatly improves performance, regaining between 19% and 54% of the

			BLEU			
Dataset	**Noise Probability**	**Noised Tokens**	**Clean Training Data**	**+ Synthetic Noise**	**Δ**	**% Recovered**
de-en	0.00%	0.00%	34.20	33.53	–0.67	–
de-en	25.00%	9.72%	27.93	31.32	3.39	54.1%
de-en	100.00%	39.36%	12.49	23.34	10.85	50.0%
fr-en	0.00%	0.00%	39.61	39.94	0.33	–
fr-en	25.00%	13.47%	30.48	34.07	3.59	39.3%
fr-en	100.00%	53.74%	11.48	19.43	7.95	28.3%
cs-en	0.00%	0.00%	27.48	27.09	–0.39	–
cs-en	25.00%	6.14%	24.31	24.91	0.60	18.9%
cs-en	100.00%	24.53%	16.64	18.91	2.27	20.9%

Table 2: Performance on the IWSLT 2016 translation task with varying rates of natural noise in the test set. **Noise Probability** is the probability of attempting to apply natural noise to a test token, while **Noised Tokens** is the fraction of tokens that were noised in practice; not every word in the vocabulary has a corresponding misspelling.

Training Noise	**BLEU**	**Δ**
No Training Noise	12.49	
+ Deletion	17.39	4.90
+ Insertion	15.00	2.51
+ Substitution	11.99	–0.50
+ Swap	14.04	1.55
All Training Noise	23.34	
− Deletion	14.96	–8.38
− Insertion	18.81	-4.53
− Substitution	20.23	–3.11
− Swap	23.07	–0.27

Table 3: Performance on IWSLT 2016 de-en test with maximal natural noise when training with one noise type (top) and three noise types (bottom).

BLEU score that was lost to natural noise. Moreover, this training regime has minimal impact on clean text translations, with negative and positive fluctuations that are smaller than 1 BLEU point.

To determine the ceiling performance of noise-based training, we split the set of natural typos and used one part for training and the other for test. However, we observed that training on natural noise behaves very similarly to training without noise at all (not shown), perhaps because the natural typos did not have enough variance to encourage the model to generalize well.

Ablation Analysis To determine the individual contribution of each type of synthetic noise, we conduct an ablation study. We first add only one type of synthetic noise at 10% (i.e. 90% of the training data is clean), and measure performance. We then take the full set of noise types, and remove a single type at each time to see how important it is given the other noises.

Table 3 shows the model's performance on the German dataset when training with various mixtures of noise. We find that deletion is by far the most effective synthetic noise in preparing our model for natural errors, followed by insertion. We observe the same trend for French and Czech. This result could explain why our experiments show a significant improvement when training on synthetic noise, while previous work, which trained only on synthetic substitutions and swaps, did not observe similar improvements.

Dataset	**Del**	**Ins**	**Sub**	**Swap**
de-en	16.6%	26.5%	17.0%	6.0%
fr-en	11.8%	11.4%	9.7%	2.6%
cs-en	6.6%	6.1%	41.7%	0.4%

Table 4: The proportion of natural errors caused by deleting/inserting/substituting a single character or swapping two adjacent characters.

Natural Noise Analysis Finally, we analyze how well our synthetic noise covers the distribution of natural noise. Table 4 shows the percentage of noised tokens that can be covered by a single noising operation. With the exception of substitutions in Czech, higher overlap between synthetic and natural noise appears to correlate with higher recovery rate in Table 2. One possible explanation for this outlier is that random synthetic substitutions might be less effective at imitating real substitutions, and that perhaps a more informed model is needed for simulating synthetic substitutions.

4 Translating Social Media Text

We also apply our synthetic noise training procedure to social media, using the recently-released MTNT dataset of Reddit posts (Michel and Neubig, 2018), focusing on the English-French translation pair. Note that no noise was inserted into the test data in this case; the only source of noise is the

Dataset	Clean Train	+ Synthetic Noise
en-fr	21.1	20.6
fr-en	23.6	24.1

Table 5: The performance of a machine translation model on the MTNT task.

non-standard spellings inherent to the dataset.

As shown in Table 5, noised training has minimal impact on performance. We did not exhaustively explore the space of possible noising strategies, and so these negative results should be taken only as a preliminary finding. Nonetheless, there are reasons to believe that synthetic training noise may not help in this case. Michel and Neubig (2018) note that the rate of spelling errors, as reported by a spell check system, is not especially high in MTNT; other differences from standard corpora include the use of entirely new words and names, terms from other languages (especially English), grammar differences, and paralinguistic phenomena such as emoticons. These findings align with prior work showing that social media does not feature high rates of misspellings (Rello and Baeza-Yates, 2012). Furthermore, many of the spelling variants in MTNT have very high edit distance (e.g., *catholique* → *catho* [Fr]). It is unlikely that training with mild synthetic noise would yield robustness to these variants, which reflect well-understood stylistic patterns rather than random variation at the character level.[1]

5 Related work

The use of noise to improve robustness in machine learning has a long history (e.g., Holmstrom and Koistinen, 1992; Wager et al., 2013), with early work by Bishop (1995) demonstrating a connection between additive noise and regularization. To achieve robustness to orthographical errors, we require noise that operates at the character level. Heigold et al. (2017) demonstrated that synthetic noising operations such as random swaps and replacements can degrade performance when inserted at test time; they also show that some robustness can be obtained by inserting the same noise at training time. Similarly, Sperber et al. (2017) explore the impact of speech-like noise.

Most relevant for us is the work of Belinkov and Bisk (2018), who evaluated on natural noise obtained from Wikipedia edit histories (e.g., Max and Wisniewski, 2010). They find that robustness to natural noise can be obtained by training on the same noise model, but that (a) training on synthetic noise does *not* yield robustness to natural noise at test time, and (b) training on natural noise significantly impairs performance on clean text. In contrast, we show that training on the right blend of synthetic noise can yield substantial improvements on natural noise at test time, without significantly impairing performance on clean data. Our ablation results suggest that deletion and insertion noise (not included by Belinkov and Bisk) are essential to achieving robustness to natural noise.

An alternative to noise infusion is to build character-level encoders that are robust to noise by design. Belinkov and Bisk (2018) experiment with a bag of characters, while Sakaguchi et al. (2017) use character-level recurrent neural networks combined with special representations for the first and last characters of each token. These models are particularly suited for specific types of swapping and scrambling noises, but are not robust to natural noise. We conducted preliminary experiments with noise-invariant encoders, but obtained better results by adding noise at training time. A related idea is to optimize an adversarial objective, in which a discriminator tries to distinguish noised and clean examples from their encoded representations (Cheng et al., 2018). This improves performance on clean data, but it makes optimization unstable, which is a well-known defect of adversarial learning (Arjovsky et al., 2017). Cheng et al. (2018) do not evaluate on natural noise.

6 Conclusion

This work takes a step towards making machine translation robust to character-level noise. We show how training on synthetic character-level noise, similar in spirit to dropout, can significantly improve a translation model's robustness to natural spelling mistakes. In particular, we find that deleting and inserting random characters play a key role in preparing the model for test-time typos. While our method works well on misspellings, it does not appear to generalize to non-standard text in social media. We conjecture that spelling mistakes constitute a small part of the deviations from standard text, and that the main challenges in this

[1] Contemporaneous work shows that MTNT performance can be improved by a domain-specific noising distribution that includes character insertions and deletions, as well as the random insertion of emoticons, stopwords, and profanity (Vaibhav et al., 2019). The specific impact of spelling noise is not evaluated, nor is the impact on clean text.

domain stem from other linguistic phenomena.

Acknowledgments Thanks to the anonymous reviewers for their feedback. We also thank Luke Zettlemoyer and our colleagues at FAIR for valuable feedback. Specifically, we thank Abdelrahman Mohamed for sharing his expertise on non-autoregressive models.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International Conference on Machine Learning*, pages 214–223.

Yonatan Belinkov and Yonatan Bisk. 2018. Synthetic and natural noise both break neural machine translation. In *ICLR*.

Chris M Bishop. 1995. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116.

Mauro Cettolo, Niehues Jan, Stüker Sebastian, Luisa Bentivogli, Roldano Cattoni, and Marcello Federico. 2016. The IWSLT 2016 evaluation campaign. In *International Workshop on Spoken Language Translation*.

Yong Cheng, Zhaopeng Tu, Fandong Meng, Junjie Zhai, and Yang Liu. 2018. Towards robust neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1766, Melbourne, Australia. Association for Computational Linguistics.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 31–36, Melbourne, Australia. Association for Computational Linguistics.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 conference of the North American Chapter of the association for computational linguistics: Human language technologies*, pages 359–369.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Georg Heigold, Günter Neumann, and Josef van Genabith. 2017. How robust are character-based word embeddings in tagging and mt against wrod scramlbing or randdm nouse? *arXiv preprint arXiv:1704.04441*.

Lasse Holmstrom and Petri Koistinen. 1992. Using additive noise in back-propagation training. *IEEE Transactions on Neural Networks*, 3(1):24–38.

Yoon Kim, Yacine Jernite, David Sontag, and Alexander M Rush. 2016. Character-aware neural language models. In *AAAI*, pages 2741–2749.

Aurélien Max and Guillaume Wisniewski. 2010. Mining naturally-occurring corrections and paraphrases from wikipedia's revision history. In *LREC*.

Paul Michel and Graham Neubig. 2018. Mtnt: A testbed for machine translation of noisy text. In *EMNLP*.

Luz Rello and Ricardo A Baeza-Yates. 2012. Social media is not that bad! the lexical quality of social media. In *ICWSM*.

Keisuke Sakaguchi, Kevin Duh, Matt Post, and Benjamin Van Durme. 2017. Robsut wrod reocginiton via semi-character recurrent neural network. In *AAAI*, pages 3281–3287.

Karel Šebesta, Zuzanna Bedřichová, Kateřina Šormová, Barbora Štindlová, Milan Hrdlička, Tereza Hrdličková, Jiří Hana, Vladimír Petkevič, Tomáš Jelínek, Svatava Škodová, Petr Janeš, Kateřina Lundáková, Hana Skoumalová, Šimon Sládek, Piotr Pierscieniak, Dagmar Toufarová, Milan Straka, Alexandr Rosen, Jakub Náplava, and Marie Poláčková. 2017. CzeSL grammatical error correction dataset (CzeSL-GEC). LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

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

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*.

Vaibhav, Sumeet Singh, Craig Stewart, and Graham Neubig. 2019. Improving robustness of machine translation with synthetic noise. *arXiv preprint arXiv:1902.09508*.

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.

Stefan Wager, Sida Wang, and Percy S Liang. 2013. Dropout training as adaptive regularization. In *Advances in neural information processing systems*, pages 351–359.

Torsten Zesch. 2012. Measuring contextual fitness using error contexts extracted from the wikipedia revision history. In *Proceedings of the 13th Conference* *of the European Chapter of the Association for Computational Linguistics*, pages 529–538. Association for Computational Linguistics.