# How to Learn in a Noisy World? *Self*-Correcting the Real-World Data Noise in Machine Translation

**Yan Meng Di Wu Christof Monz**
Language Technology Lab
University of Amsterdam
{y.meng}@uva.nl

## Abstract

The massive amounts of web-mined parallel data often contain large amounts of noise. Semantic misalignment, as the primary source of the noise, poses a challenge for training machine translation systems. In this paper, we first introduce a process for simulating misalignment controlled by semantic similarity, which closely resembles misaligned sentences in real-world web-crawled corpora. Under our simulated misalignment noise settings, we quantitatively analyze its impact on machine translation and demonstrate the limited effectiveness of widely used pre-filters for noise detection. This underscores the necessity of more fine-grained ways to handle hard-to-detect misalignment noise. By analyzing the reliability of the model's self-knowledge for distinguishing misaligned and clean data at the token level, we propose *self-correction*—an approach that gradually increases trust in the model's self-knowledge to correct the supervision signal during training. Comprehensive experiments show that our method significantly improves translation performance both in the presence of simulated misalignment noise and when applied to real-world, noisy web-mined datasets, across a range of translation tasks.

## 1 Introduction

The success of machine translation (MT) models is mainly due to the availability of large amounts of web-crawled parallel data. However, publicly available web-mined parallel corpora such as CCAligned (El-Kishky et al., 2020), WikiMatrix (Schwenk and Douze, 2017) and ParaCrawl (Bañón et al., 2020) are shown to be noisy (Kreutzer et al., 2022; Ranathunga et al., 2024). The notable performance drop in NMT quality when training with injected synthetic noise (Khayrallah and Koehn, 2018) or fine-tuning with CCAligned (Lee et al., 2022) indicates the importance of improving the model's robustness when training on a noisy corpus.

Given a noisy training dataset, a common and straightforward approach to mitigate the impact of noisy data is to filter low-quality training samples (Herold et al., 2022; Bane et al., 2022). However, in practice, large amounts of misalignments still exist in *pre-filtered* web-mined datasets (Kreutzer et al., 2022). This is because real-world misaligned sentences often share partial meanings, making them appear as seemingly parallel, increasing the difficulty for pre-filters to detect them. To quantitatively analyze such hard-to-detect real-world misalignments, we design a process to simulate it controlled by semantic similarity. Unlike earlier works (Khayrallah and Koehn, 2018; Herold et al., 2022; Li et al., 2024) that generate misaligned bitext by random shuffling—an approach that is both unrealistic and easy to detect—our simulated misalignments closely resemble real-world noise and challenge widely-used pre-filters, such as LASER (Artetxe and Schwenk, 2018) and COMET (Rei et al., 2020).

Under our simulated noise settings, we evaluate a type of approach that could potentially handle misalignment noise: Data truncation (Kang and Hashimoto, 2020; Li et al., 2024; Flores and Cohan, 2024), which ignores losses at the token level during training when there is a relatively large discrepancy between the model's prediction and the ground truth. Although promising, we observe that truncation methods are sensitive to varying levels of misalignment noise. For example, for low-resource corpora with a high misalignment rate, truncation methods even *degrade* the translation performance; see Section 5.3. We argue that the noisy low-resource setting prevents the model from acquiring sufficient correct knowledge, resulting in an inaccurate removal of clean and useful ground-truth data. Moreover, truncation methods start to ignore potential data noise from an early training time, which overlooks the increasing reliability of the model's prediction over time.



*Findings of the Association for Computational Linguistics: NAACL 2025*, pages 7466–7482
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

To overcome these limitations, we propose an approach called *self-correction*, which leverages the model's self-knowledge to correct noise during training while maintaining supervision from the ground truth to avoid discarding useful training information. To adapt to the model's changing reliability, we set a dynamic schedule to gradually increase trust in its output. During the early stages of training, we place greater trust in the reference over the model's predictions. As the model acquires more knowledge, we progressively use the model's predictions to revise the ground truth.

We evaluate our self-correction method in both simulated and real-world noisy settings. We demonstrate that our method consistently outperforms baselines in both high- and low-resource datasets with different levels of misalignment noise. Moreover, we clearly show that gains are mainly due to revising the misaligned samples while maintaining the performance of clean parallel data. In the real-world noise setting, our self-correction method effectively handles naturally occuring noise in web-mined parallel datasets, e.g., ParaCrawl and CCAligned, achieving performance gains of up to 2.1 BLEU points across seven translation tasks and outperforming alternative methods, including pre-filters and truncation.

# 2 Background

## 2.1 The Noisy World

Web-crawled parallel corpora are the primary training data source for machine translation models. However, parallel data crawled from public websites lack quality guarantees and contain different types of noise (Kreutzer et al., 2022), including wrong language, non-linguistic content, and semantic misalignment.

The primary source of noise in parallel web-mined data is semantic misalignment (Khayrallah and Koehn, 2018; Kreutzer et al., 2022; Ranathunga et al., 2024). For instance, Khayrallah and Koehn (2018) analyzed the data quality of the raw ParaCrawl corpus, showing 77% of the analyzed sentence pairs to contain noise with half of them being misalignments. Wrong language and non-linguistic contents only account for a small portion and can be easily handled by filters, e.g., language identification toolkits (Herold et al., 2022). Kreutzer et al. (2022) extended the data quality analysis to pre-filtered web-mined datasets, e.g., WikiMatrix, CCAligned, noting that more than 50% of data in both corpora are noisy with misalignments being the primary reason.

Overall, previous studies demonstrate the prevalence of noisy training data in web-mined corpora for machine translation and underscore the importance of noise-robust training, particularly in handling misaligned data.

## 2.2 Learning in the Noisy World

### 2.2.1 Data Filter

Data filtering is a straightforward way to mitigate the impact of noise from translation corpora. Two types of filters are often used to ensure semantic alignment in a sentence pair: (1) surface-level filters, e.g., removing sentence pairs that differ a lot in source and target length; (2) semantic-level filters, relying on quality estimation models to score each sentence pair (Kepler et al., 2019; Rei et al., 2020; Peter et al., 2023). Other works consider misalignment detection as a ranking problem by training a classifier on annotated synthetic misaligned data (Briakou and Carpuat, 2020).

In this paper, we mainly consider semantic-level filters for comparison, e.g., LASER (Artetxe and Schwenk, 2018) and COMET (Rei et al., 2020), due to their broad applicability and common usage.

### 2.2.2 Training Robustness

The primary limitation of data filters is that they discard entire training samples before training. To retain as much useful information as possible in noisy samples, several methods focus on mitigating their negative impact during model training. For instance, Wang et al. (2018) propose an online data selection approach that utilizes extrinsic trusted data to identify high-quality samples during training. Similarly, Briakou and Carpuat (2021) employ external semantic divergence tags to guide the training of the translation model. However, both of these approaches depend on external data or factors.

In this paper, we consider an alternative line of works, i.e., data truncation, which relies solely on the model's self-knowledge to ignore potential noise and further benefits the robustness of model training (Kang and Hashimoto, 2020; Li et al., 2024). For example, Kang and Hashimoto (2020) use losses to estimate data quality, where tokens with high loss are considered as noise and will be ignored during training by setting their loss to zero. Li et al. (2024) further propose Error Norm Truncation, using the $l_2$ norm between the model's

| | |
|---|---|
| en | **Alcohol poisoning** is the biggest cause of death. |
| nl | **Jacht** is de belangriekste doodsoorzaak.<br>en: ***Hunting*** *is the biggest cause of death.* |
| en | With Bravofly you can compare the flight prices Santa Cruz De La Palma of **over 400 of the most famous** airlines **in the world.** |
| de | Bravofly findet für Sie sämtliche **Billigflüge Zürich** - Santa Cruz De La Palma der besten **europäischen** Billigfluggesellschaften.<br>en: *Bravofly finds all the* ***cheap*** *flights* ***Zurich*** *- Santa Cruz De La Palma from the best* ***European low-cost*** *airlines for you.* |

Table 1: Examples of misaligned sentences in the ParaCrawl dataset. **Bold** represents the misaligned meanings. *Italic* text represents the English translation.

| Misaligned Types | Adequacy |
|---|---|
| Real-World | 3.1 |
| Misaligned-COMET | 2.7 |
| Misaligned-LASER | 2.6 |
| Misaligned-Random | 1.2 |

Table 2: Adequacy (scale: 1–5) scores on simulated and real-world misaligned sentences. The real-world misaligned sentences are selected from ParaCrawl V7.0. Misaligned-COMET/LASER and real-world misaligned targets convey partial meanings with the sources.

prediction distribution and the one-hot ground-truth token distribution to measure data quality. Their method considers the model's prediction distribution of non-target tokens, providing a more accurate data quality measurement.

However, there are two limitations of truncation methods: First, they ignore the potential noisy training tokens from a specific training iteration, which overlooks the changes in the model's reliability during training. Second, ignoring can remove partially clean training information, which can be harmful for low-resource tasks. In this paper, we go a step further and propose a self-correction method to gradually increase the trust of model prediction distributions to correct rather than ignore the ground-truth data during training. Details are introduced in Section 4.

## 3 An Empirical Study of Misalignment

In this section, we investigate the primary source of noise, i.e., semantic misalignment, in a simulated setting. We first introduce a strategy to simulate realistic misalignment noise by controlling semantic similarity (Section 3.1). Next, we show the similarity of our simulated noise to real-world misalignment in terms of adequacy and its hard-to-detect nature (Section 3.2). Under our simulated noisy setting, we evaluate model-based metrics to distinguish data noise and highlight their potential limitations (Section 3.3).

### 3.1 Simulating Misalignment Noise

To simulate misalignment, previous works (Bane et al., 2022; Herold et al., 2022; Li et al., 2024) randomly shuffle target sentences of a clean parallel corpus. However, random shuffling noise can be easily removed by pre-filters based on length or obvious semantic differences (Herold et al., 2022), oversimplifying misalignments found in real-world web-mined corpora. While Briakou and Carpuat (2020) proposed generating fine-grained misaligned targets by perturbing equivalent samples, e.g., deletion or replacement, their method does not guarantee the fluency and authenticity of the misaligned sentences.

To quantitatively analyze the impact of realistic misalignment noise, we designed a process to simulate real-world misalignment controlled by semantic similarity. The main idea is to select misaligned target sentences from a large pool of clean candidates that share partial semantics with the corresponding source sentences, where we use semantic-level models, e.g., LASER or COMET, to measure semantic similarity across languages.

More specifically, given a source sentence and a large pool of target sentences, we first narrow down potential candidates based on the length differences and the word overlap ratio with the true parallel target to reduce computational costs. Then, the candidate with the highest semantic similarity score is selected as the final synthetic misaligned target. By this two-step process, we generate misalignment efficiently while maintaining shared semantics. Algorithm 1 provides a detailed description of our strategy. Examples of misaligned sentences generated using LASER (Misaligned-LASER) and COMET (Misaligned-COMET) can be found in Appendix 6.

### 3.2 Real-World Misalignment

#### 3.2.1 Adequacy

To show the similarity of our simulated noise to real-world misalignment, we conduct a human evaluation of 200 simulated and real-world misaligned sentences, rating their *Adequacy* (scale 1–5), which measures the meaning overlap between source and target. In Table 2, we show that both real-world misalignment and Misaligned-

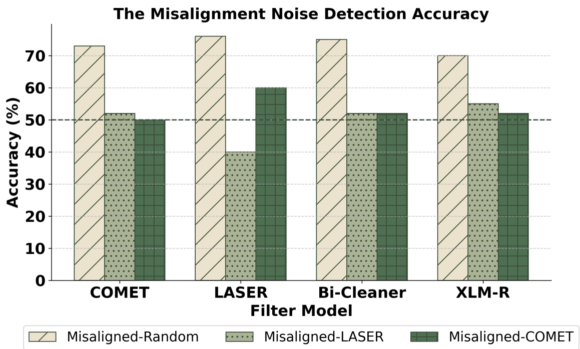


Figure 1: The accuracy of various data filters in distinguishing misaligned noise from clean parallel data. All four data filters **perform similarly to random guessing** (indicated by the black dashed line) on Misaligned-LASER/COMET.

LASER/COMET (see Section 3.1) have a relatively high adequacy score, above 2.5, while random shuffled misaligned sentences only have an adequacy of 1.2. This ensures our simulated misalignment contains only partial semantic overlaps as the real-world misalignment. Details of the human evaluation are in Appendix B.3.

#### 3.2.2 Hard-to-Detect Nature

To show the hard-to-detect nature of our simulated noise, we investigate the noise detection ability of widely used pre-filters: COMET, LASER, Bi-Cleaner, and XLM-R. The details for each filter model are provided in Appendix A.

We calculate the noise detection accuracy of the data filters on a mixed set with the same amounts of clean and noisy data. For the clean data, we randomly sample 2,000 clean sentence pairs from the WMT2017 De→En test set. For Misaligned-Random, we randomly shuffle the order of target sentences in the sampled clean sentence pairs. For Misaligned-COMET and Misaligned-LASER, we use the same source sentences from the sampled clean data. We select the misaligned targets from another 200K target sentences in the training corpus based on Algorithm 1. We score each sentence pair based on the filter models and determine a true ratio threshold based on the amounts of clean and noisy sentence pairs, here 1:1. Sentence pairs with scores below this threshold are classified as noisy.

Figure 1 shows the noise detection accuracy of the data filters for different misaligned noise. First, all data filters have a relatively high detection accuracy for Misaligned-Random, particularly when using LASER, with an accuracy of 76%. This challenges previous assumptions (Khayrallah and Koehn, 2018; Li et al., 2024) of the impact of misalignment noise on translation performance since most of them can be pre-filtered. However, our introduced noise, i.e., Misaligned-LASER and Misaligned-COMET, presents difficulties for all pre-filters, as real-world misalignments do.

Overall, we show the validity of our simulated noise in two aspects: (1) Adequacy, reflected in the similar level of shared semantics as real-world misalignments; (2) Hard-to-Detect Nature, reflected in the low noise detection accuracy from widely used pre-filters.

### 3.3 Fine-grained Misalignment Detection

To measure data quality during training, token-level loss and error norm values are used in data truncation methods (Kang and Hashimoto, 2020; Li et al., 2024). Here, we evaluate their effectiveness under our simulated misalignment noise settings.

Loss measures the model's predicted probability of the ground-truth token. On the other hand, error norm value (*el2n*) calculates the difference between the ground-truth (one-hot) distribution $OH(y_t)$ and the model's prediction distribution $p_\theta(\cdot|y_{<t}, x)$ (eq 1). Tokens with relatively high *loss* or *el2n* values are indicated as noise.

$$el2n = ||p_\theta(\cdot|x, y_{<t}) - OH(y_t)||_2. \quad (1)$$

We record the *loss* and *el2n* values for each token from 2,000 clean and Misaligned-LASER target sentences in the same data setting as in Section 3.2.2. Figure 2 shows that clean and misaligned sentences have different *loss* and *el2n* distributions as training time increases from epoch 5 to 30. This shows the effectiveness of the model's self-knowledge for distinguishing hard-to-detect misalignment noise from clean sentences.

Notably, the *el2n* metric exhibits stronger differentiability compared to *loss*, underscoring the importance of considering the model's full prediction distribution. However, the noisy samples' *el2n* distribution still partially shifts towards lower values during training, mainly due to the presence of clean tokens in the simulated misaligned sentences. To confirm that the shifted tokens in the noisy samples are truly clean, we provide token-level annotations (see Appendix B.4) to show that annotated misaligned tokens do have higher *el2n* values (avg. 1.13) than clean ones (avg. 0.32).

Interestingly, we also observe that clean samples contain tokens with high *el2n* values (see Table 9). We hypothesize that these tokens might be difficult

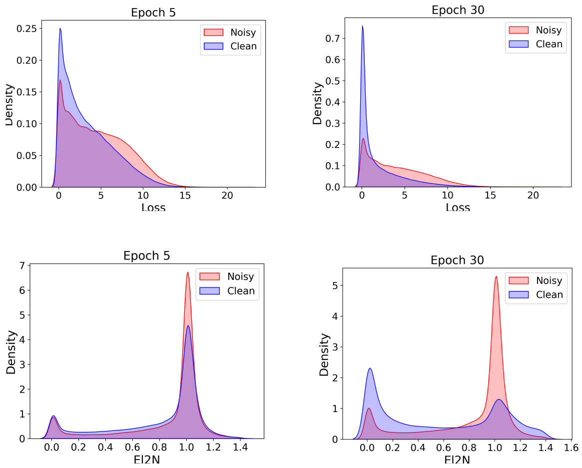


Figure 2: *loss* (above) and *el2n* (below) distribution for clean and Misaligned-LASER noise samples during the training process (Epoch = 5 and 30). Red distribution represents misaligned-*LASER* noise and Blue distribution represents the clean data. **As training progresses, *el2n* distributions for clean and noisy data shift differently.** The distribution plots for the full training process are in the Appendix in Figure 4.

for the model to learn. Future work could further differentiate between hard-to-learn and noisy tokens and explore their respective impacts on the model's performance.

Overall, we point out two limitations of truncation methods relying on model-based metrics: First, they overlook the increasing reliability of model predictions by removing potential data noise already during early training stages. Second, they cannot avoid ignoring clean but useful data. As mentioned, partial clean tokens still have high *el2n* values.

## 4 Noise *Self*-Correction

To overcome the limitations of truncation methods in Section 3.3, we propose a self-correction method to gradually increase the trust of the model's prediction distributions to correct the supervision during training. Our method keeps the supervision signals from the training data to avoid clean training information loss and also progressively trusts a dynamic entropy state of the model's prediction to revise the data. Our work is in line with label correction in computer vision (discussed in Appendix C.1).

**New Target.** Consider conditional probability models $p_\theta(y|x)$ for machine translation. Such models assign probabilities to a target sequence $y = (y_1, ..., y_T)$ by factorizing it to the sum of log probabilities of individual tokens $y_i$ from vocabulary $V$. At each training iteration, the model learns towards the ground-truth token distribution, one-hot $q(y_i)$, with a model prediction distribution $p_\theta(\cdot|x, y_{<i})$. In self-correction, we leverage the model prediction $p_\theta(\cdot|x, y_{<i})$ to revise the one-hot distribution $q(y_i)$ with the aim of learning towards a new target $\bar{q}(y_i)$:

$$\bar{q}(y_i) = (1 - \lambda)q(y_i) + \lambda p_\theta(\cdot|x, y_{<i}) \qquad (2)$$

In this way, the new target $\bar{q}(y_i)$ keeps the original supervision signal from the training data and the model's prediction. $\lambda$ denotes a weighting factor that determines how much to trust the model prediction.

**Dynamic Learning Schedule.** We correlate $\lambda$ with a learning time function $\mathrm{Time}(t)$ of training iteration $t$ and model entropy $H(p_\theta)$:

$$\lambda = (1 - H(p_\theta)) \times \mathrm{Time}(t) \qquad (3)$$

For $H(p_\theta)$, the model trusts its prediction more when it has a more confident prediction, i.e., lower entropy. For $\mathrm{Time}(t)$, the model can trust its self-knowledge as training progresses. We use a schedule (Bengio et al., 2015) to increase $\mathrm{Time}(t)$ as a function of the training iteration $t$ and $T$ as the number of total iterations.

$$\mathrm{Time}(t) = \frac{1}{1 + \exp(\beta(\frac{t}{T} + \alpha))} \qquad (4)$$

where $\alpha$ and $\beta$ are hyper-parameters[1].

In general, at the beginning of training, the model is not well-trained, and a small $\mathrm{Time}(t)$ value controls the model to rely more on the ground-truth data than its own predictions. As training progresses, increasing $\mathrm{Time}(t)$ allows the model to trust more in its reliable prediction.

**Sharpen the Model Prediction.** To overcome the overly uncertain model prediction when learning towards the new target in Equation 2, we sharpen the model prediction distribution by controlling the softmax temperature $\tau$ in $\bar{p}_\theta = \frac{\exp(z_i/\tau)}{\sum_{j=1}^{N}\exp(z_j/\tau)}$. We control $\tau$ in a dynamic way to vary it inversely with $\mathrm{Time}(t)$. Therefore, $\tau$ gradually decreases as training goes on: a higher $\tau$ value at early training stages can prevent the model from converging and a smaller $\tau$ in the later stage makes the model more confident in its output.

[1]We choose $\alpha$ and $\beta$ based on prior experiments, see Appendix C.2.

| | | Misaligned-LASER | | | Misaligned-COMET | | | Raw-Crawl Data | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10% | 30% | 50% | 10% | 30% | 50% | 10% | 30% | 50% |
| **Baseline** | *with* noise | 33.0$^*$ | 31.7$^*$ | 30.5$^*$ | 33.1$^*$ | 32.0$^*$ | 30.0$^*$ | 33.0$^*$ | 31.5$^*$ | 29.6$^*$ |
| **Oracle** | *w/o* noise | 33.3 | 32.7 | 32.0 | 33.3 | 32.7 | 32.0 | 33.3 | 32.7 | 32.0 |
| **Pre-Filter** | LASER | 33.2 | 31.4$^*$ | 30.0$^*$ | 33.1$^*$ | **32.6** | 30.2$^*$ | 33.0$^*$ | 31.6$^*$ | 30.0$^*$ |
| | COMET | 32.9$^*$ | 31.5$^*$ | 30.4$^*$ | 33.0$^*$ | 31.7$^*$ | 29.6$^*$ | 32.4$^*$ | 31.6$^*$ | 28.5$^*$ |
| **Truncation** | *loss* | 33.1$^*$ | 31.4$^*$ | 30.7$^*$ | 33.0$^*$ | 31.2$^*$ | 29.8$^*$ | 33.0$^*$ | 31.8 | 29.9$^*$ |
| | *el2n* | 33.0$^*$ | 31.9$^*$ | 31.0$^*$ | 32.9$^*$ | 31.8$^*$ | 29.9$^*$ | 33.0$^*$ | 31.6$^*$ | 30.0$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | 33.1 | **32.9** | 31.3 | 33.2 | 32.4 | 30.4 | 33.4 | 31.7 | 30.3 |
| | dynamic $\tau$ | **33.5** | 32.3 | **31.4** | **33.3** | 32.5 | **30.6** | **33.5** | **31.9** | **30.4** |

Table 3: SacreBLEU scores of high-resource De $\rightarrow$ En translation task with different types of noise. The BLEU score of the full clean training corpus (5.8M) De $\rightarrow$ En is 33.5. **Baseline** *with* noise: represents the translation performance when injecting with 10%, 30%, 50% of data noise. **Oracle** *w/o* noise: represents the upper-bound translation performance when training with the remaining clean data, specifically 90%, 70%, 50% of the data excluding the noise. **Bold** and Underline represents the best and second best score. $*$ signifies that our self-correction method (dynamic $\tau$) is significantly better (p-value < 0.05) than the comparing methods. The statistical significance results with paired bootstrap resampling are followed by (Koehn, 2004). COMET and Chrf++ scores are provided in Table 12 in Appendix E.

In Section 5, we compare the performance of both fixed[2] and dynamic $\tau$ to self-correct the data noise and also show the impact of different values of fixed $\tau$ on the performance in Appendix C.3.

**Training.** After acquiring a new target $\bar{q}(y_i)$, derived from both the ground truth and the model's own predictions, we obtain a new training objective based on maximum likelihood estimation (MLE). The following loss function is minimized for every training token over the training corpus $D$:

$$L_\theta(x, y) = \mathbb{E}_{y_i \sim D} \left[ -\bar{q}(y_i) \log p_\theta(\cdot | x, y_{<i}) \right] \quad (5)$$

## 5 Experiments

In this section, we investigate the effectiveness of our self-correction method for translation tasks in two experimental settings: simulated and real-world noisy settings. For the simulated noisy setting (Section 5.2), we conduct experiments by injecting two types of noise, raw-crawl data and simulated misaligned noise, into a clean translation corpus. For the real-world noisy setting (Section 5.3), we perform experiments on two noisy web-mined datasets, i.e., ParaCrawl and CCAligned, across different language pairs.

### 5.1 Comparing Systems

We compare our self-correction method with the following comparing systems:[3]

**Pre-Filtering.** We select two widely used data filters: LASER and COMET. We rank the training sentence pairs based on the scores calculated by the filter models. For the simulated noise experiments (Section 5.2), we filter out the sentence pairs with the lowest scores before training, matching the size to the injected data noise. The training data size for pre-filter methods is 90%, 70%, and 50% of the full training corpus when injecting with 10%, 30%, and 50% of data noise. For the real-world noise experiments (Section 5.3), we filter out 20% of the sentence pairs with the lowest scores.

**Truncation.** We compare two truncation methods: (1) *loss* truncation (Kang and Hashimoto, 2020), (2) error norm value (*el2n*) truncation (Li et al., 2024). Following (Li et al., 2024), we choose the best result among three truncation fractions $\{0.05, 0.1, 0.2\}$ for both *loss* and *el2n* truncation. The starting iteration to truncate data is set as 1,500.

### 5.2 Simulated Noisy World

#### 5.2.1 Experimental Setup

We conduct experiments on both high- and low-resource translation tasks. We use the WMT2017 (German) De$\rightarrow$En news translation data as the high-resource task and En$\rightarrow$Si (Sinhala) from OPUS[4] as the low-resource task.

Following Herold et al. (2022), we inject noise by replacing a portion (10%, 30%, 50%) of the clean training corpus with simulated misalignment noise or raw crawl data. The misalignment noise is generated by Algorithm 1 from the replaced portion of the clean corpus. The raw crawl data noise is randomly selected from the raw Paracrawl corpus[5]. Specifically, the raw crawl data provides a realistic test bed for noise-handling methods since

[2] We use fixed $\tau = 0.5$ followed by (Wang et al., 2022).

[3] Note that all the models' details align with the corresponding baselines.

[4] https://opus.nlpl.eu/

[5] https://paracrawl.eu/

| | | Misaligned-LASER | | | Misaligned-COMET | | | Raw-Crawl Data | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 10% | 30% | 50% | 10% | 30% | 50% | 10% | 30% | 50% |
| **Baseline** | *with* noise | 22.3 | 20.0$^*$ | 18.0$^*$ | 21.4$^*$ | 18.7$^*$ | 14.2$^*$ | 22.3 | 21.0$^*$ | 19.0$^*$ |
| **Oracle** | *w/o* noise | 22.3 | 21.0 | 20.8 | 22.3 | 21.0 | 20.8 | 22.3 | 21.0 | 20.8 |
| **Pre-Filter** | LASER | 22.0$^*$ | 18.7$^*$ | 17.0$^*$ | 21.1$^*$ | 18.9$^*$ | **16.3** | 21.0$^*$ | 21.2$^*$ | 19.2$^*$ |
| | COMET | 22.0$^*$ | 20.0$^*$ | 17.6$^*$ | 21.0$^*$ | 18.6$^*$ | 13.8$^*$ | 22.2 | 20.9$^*$ | 18.9$^*$ |
| **Truncation** | *loss* | 22.1 | 20.5 | 17.9$^*$ | 20.0$^*$ | 17.2$^*$ | 14.2$^*$ | 22.2 | 21.1$^*$ | 19.1$^*$ |
| | *el2n* | 22.0$^*$ | 20.5 | 18.2$^*$ | 21.1$^*$ | 18.9$^*$ | 14.3$^*$ | 22.0$^*$ | 21.3$^*$ | 19.2$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | **22.4** | **21.2** | <u>19.8</u> | <u>21.7</u> | <u>19.0</u> | 15.3 | **22.5** | <u>21.5</u> | **19.9** |
| | dynamic $\tau$ | <u>22.3</u> | <u>20.7</u> | **20.2** | **22.1** | **19.6** | <u>16.2</u> | <u>22.3</u> | **21.9** | <u>19.6</u> |

Table 4: SacreBLEU scores of low-resource En $\rightarrow$ Si translation task with different types of noise. The BLEU score of full clean training corpus (0.9M) En $\rightarrow$ Si is 22.5. Chrf++ and COMET score are provided in Table 13 in Appendix E.

it contains a mixture of naturally occurring noise, including misaligned sentences, wrong language, grammar errors, etc.

All translation models use the fairseq (Ott et al., 2019) implementation of the Transformer-Big architecture for the high-resource task and Transformer-Base for the low-resource task. The full training details are shown in Appendix D.1.

### 5.2.2 Results

Tables 3 and 4 show the high-resource De$\rightarrow$En and low-resource En$\rightarrow$Si translation performance trained on the corpus with simulated misalignment or raw crawl data noise. Overall, both noise settings negatively impact translation quality, as shown by the performance drop with increasing noise levels.

First, we show that pre-filter COMET fails to filter Misaligned-LASER noise, leading to a drop in translation performance in both high-resource and low-resource scenarios. This finding aligns with Bane et al. (2022), which demonstrates that COMET is weak at detecting misaligned segments. On the other hand, pre-filter LASER is effective in handling Misaligned-COMET noise but only achieves modest gains when dealing with raw-crawl data noise.

Second, we demonstrate the effectiveness of leveraging the model's self-knowledge to detect data noise during training. Consistent with our findings in Section 3.3, we show that using the *el2n* metric yields better performance compared to using *loss*. However, *el2n* truncation still falls short in highly noisy environments (50%). In such cases, the noisy datasets prevent the model from acquiring accurate knowledge, leading to incorrect data removal during training.

Our self-correction method overcomes the limitations of *el2n* truncation by 'revising' rather than 'ignoring' data noise. This approach retains ground truth supervision, preventing the loss of clean data information. This advantage is reflected in the superior performance of self-correction across low- and high-resource tasks in various noise settings. For instance, when injecting 50% Misaligned-LASER noise into the En$\rightarrow$Si task, our self-correction method outperforms *el2n* truncation by 2.0 BLEU points.

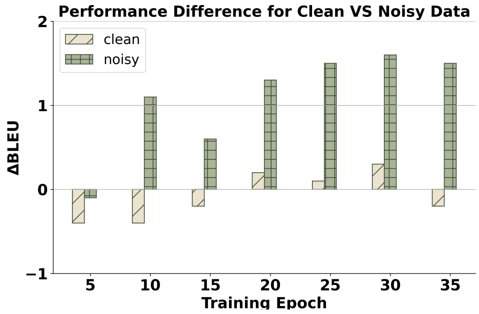


Figure 3: Performance differences between our self-correction method and baseline on noisy (Misaligned-LASER) and clean data for De$\rightarrow$En task with 30% injected misaligned-LASER. **The effectiveness of our method mainly arises from improving the misaligned noisy data over clean ones.**

Overall, our findings highlight the importance of utilizing the model's own predictions. This supports the hypothesis that training models solely on reference translations can limit performance, particularly when the reference is inferior to the model-generated translation (Xu et al., 2024).

### 5.2.3 The Sources of Improvements

The previous section shows the benefits of our self-correction method in the presence of simulated misalignment noise. To further investigate whether the improvements arise from addressing the misaligned data, we compare the differences in translation performance on clean and Misaligned-LASER data after applying the self-correction method.

Specifically, we sample 1K clean and Misaligned-LASER sentence pairs and report their BLEU score differences between the

| | | en→fr♡ | en→tr† | en→es† | en→be† | en→si♡ | en→sw♡ | en→km♡ | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| **Misaligned Rate (%)** | | 10% | 44% | 22% | 10% | 62% | 11% | 18% | - |
| **Corpus Size (M)** | | 5M | 5M | 5M | 1.1M | 210K | 130K | 60K | - |
| **Baseline** | | 41.1$^*$ | 23.5$^*$ | 21.6$^*$ | 9.9$^*$ | 7.0$^*$ | 13.0$^*$ | 4.2$^*$ | 17.1 |
| **Pre-Filter** | LASER | 41.8$^*$ | 23.2$^*$ | <u>22.5</u> | 9.8$^*$ | 6.6$^*$ | 12.7$^*$ | 3.8$^*$ | 17.2 |
| | COMET | 41.6$^*$ | 23.7$^*$ | 22.2$^*$ | 9.6$^*$ | 6.8$^*$ | 12.5$^*$ | 4.0$^*$ | 17.2 |
| **Truncation** | *loss* | 41.2$^*$ | 23.8$^*$ | 21.9$^*$ | 9.8$^*$ | 6.0$^*$ | 12.5$^*$ | 4.0$^*$ | 17.0 |
| | *el2n* | 41.3$^*$ | <u>23.9</u>$^*$ | 22.0$^*$ | 10.0$^*$ | 6.0$^*$ | 13.0$^*$ | 4.5$^*$ | 17.2 |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | <u>41.9</u> | 23.4 | 21.9 | <u>10.1</u> | <u>7.6</u> | <u>14.7</u> | <u>4.6</u> | <u>17.9</u> |
| | dynamic $\tau$ | **42.3** | **24.2** | **22.8** | **10.5** | **7.8** | **15.1** | **5.0** | **18.2** |

Table 5: SacreBLEU scores on real-world web-mined corpora. **Bold** and <u>Underline</u> represents the best and second best score. † denotes language pairs from CCAligned V1.0. ♡ denotes language pairs from ParaCrawl V7.1. $*$ indicates that our self-correction method is significantly better (p-value < 0.05) than the baseline. The misaligned noise rate for different language pairs is reported from Kreutzer et al. (2022). Chrf++ and COMET scores are provided in Table 14 in Appendix E.

baseline and the self-correction model during training. For Misaligned-LASER noisy data, BLEU scores are computed using the original parallel true references. Figure 3 shows that the effectiveness of our self-correction method primarily stems from improving the translation quality of misaligned data. Our method enhances performance on misaligned noisy data by up to 1.5 BLEU points during training, while its impact on clean data remains minimal.

## 5.3 Real Noisy World

### 5.3.1 Experimental Setup

We investigate two noisy web-crawled datasets: Paracrawl V7.1 and CCAligned V1.0. These two datasets exhibit varying semantic misalignment rates across different low- and high-resource language pairs (Kreutzer et al., 2022). For each dataset, we select language pairs with varying levels of misalignment noise rates, from high- to low-resource. Training data details for the selected language pairs are shown in Appendix D.2.2. The validation and test sets for all tasks are from Flores101[6]. We train for all tasks on the Transformer-Big (Vaswani et al., 2017) architecture.

### 5.3.2 Results

Table 5 shows the translation performance for two noisy web-crawled datasets, CCAligned V1.0 and Paracrawl V7.1, across language pairs with varying corpus size and misaligned rates.

Similar to our findings under the simulated noise setting in Section 5.2, we show that pre-filters and data truncation methods are limited to low-resource tasks with varying misalignment rates, e.g., en→sw, en→si, and en→km, even degrading the translation performance. These two methods handle data noise by removing or ignoring it; however, the noisy examples might still be partially helpful for the model, especially in data-scarce scenarios.

In contrast, the self-correction method consistently outperforms alternative methods, including pre-filters and truncation, with an overall improvement of 1.1 BLEU, 1.7 COMET, and 1.5 ChrF++ points over the baseline. Specifically, self-correction shows superior performance in low-resource tasks, with up to 2.1 BLEU and 2.4 COMET points over the baseline for en→sw task. This further emphasizes the effectiveness of using the model's self-knowledge to "correct" noise in real-world web-mined datasets.

# 6 Conclusion

In this paper, we aim to address the data quality issue in the web-mined translation corpora. We show that the primary noise source in translation corpora, namely semantic misalignment, is hard to filter or handle by both widely used pre-filtering and data truncation methods. To quantitatively analyze the impact of misalignment noise, we propose a process to simulate it controlled by semantic similarity, which reflects the partially shared meanings often found in misaligned sentence pairs from real-world web-crawled corpora.

Under our simulated misalignment noise setting, we observe increasing reliability of the model's self-knowledge for detecting misalignments at the token level. Building on this, we propose *self-correction*, which focuses on the model's training dynamics and revises the training supervision from the reference data by the model's prediction. Comprehensive experiments demonstrate the effectiveness of our approach on both simulated and real-

[6] https://github.com/facebookresearch/flores

world web-mined translation corpora.This performance outperforms alternative methods, including pre-filtering and truncation methods. Moreover, we show that the gains are mainly from revising the misaligned samples while maintaining the performance on clean data. Overall, our work provides a critical finding on the effectiveness of leveraging the model's predictions instead of solely relying on flawed reference data.

## 7 Limitation

First, we acknowledge the potential bias in our self-correction method, which could learn towards the noise due to its reliance on ground truth during the early training stages. However, we believe this is not a significant issue because our method consistently demonstrates robust experimental results across different noise scenarios. Future work could explore modifications to mitigate this potential bias and enhance performance in diverse settings.

Second, our work aims at learning from a noisy training corpus, which might limit improvements when using high-quality training datasets. Furthermore, the self-correction approach has shown promise for machine translation tasks, but another limitation is the unexplored potential for other natural language processing tasks, e.g., summarization or text generation. Future work should investigate the effectiveness of this approach across different downstream tasks.

## Acknowledgments

This research was funded in part by the Netherlands Organization for Scientific Research (NWO) under project numbers VI.C.192.080 and 2023.017. We would like to thank Vlad Niculae, David Stap, Sergey Troshin and Evgeniia Tokarchuk for their useful suggestions. We would also like to thank the reviewers for their feedback.

## References

Mikel Artetxe and Holger Schwenk. 2018. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

Fred Bane, Celia Soler Uguet, Wiktor Stribiżew, and Anna Zaretskaya. 2022. A comparison of data filtering methods for neural machine translation. In *Proceedings of the 15th Biennial Conference of the Association for Machine Translation in the Americas (Volume 2: Users and Providers Track and Government Track)*, pages 313–325, Orlando, USA. Association for Machine Translation in the Americas.

Marta Bañón, Pinzhen Chen, Barry Haddow, Kenneth Heafield, Hieu Hoang, Miquel Esplà-Gomis, Mikel L. Forcada, Amir Kamran, Faheem Kirefu, Philipp Koehn, Sergio Ortiz Rojas, Leopoldo Pla Sempere, Gema Ramírez-Sánchez, Elsa Sarrías, Marek Strelec, Brian Thompson, William Waites, Dion Wiggins, and Jaume Zaragoza. 2020. ParaCrawl: Web-scale acquisition of parallel corpora. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4555–4567, Online. Association for Computational Linguistics.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Eleftheria Briakou and Marine Carpuat. 2020. Detecting Fine-Grained Cross-Lingual Semantic Divergences without Supervision by Learning to Rank. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1563–1580, Online. Association for Computational Linguistics.

Eleftheria Briakou and Marine Carpuat. 2021. Beyond noise: Mitigating the impact of fine-grained semantic divergences on neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 7236–7249, Online. Association for Computational Linguistics.

Vishrav Chaudhary, Y. Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *ArXiv*, abs/1906.08885.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. In *Annual Meeting of the Association for Computational Linguistics*.

Ahmed El-Kishky, Vishrav Chaudhary, Francisco Guzmán, and Philipp Koehn. 2020. CCAligned: A massive collection of cross-lingual web-document pairs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5960–5969, Online. Association for Computational Linguistics.

Lorenzo Jaime Flores and Arman Cohan. 2024. On the benefits of fine-grained loss truncation: A case study on factuality in summarization. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 138–150, St. Julian's, Malta. Association for Computational Linguistics.

Christian Herold, Jan Rosendahl, Joris Vanvinckenroye, and Hermann Ney. 2022. Detecting various types of noise for neural machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2542–2551, Dublin, Ireland. Association for Computational Linguistics.

Daniel Kang and Tatsunori B. Hashimoto. 2020. Improved natural language generation via loss truncation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 718–731, Online. Association for Computational Linguistics.

Fabio Kepler, Jonay Trénous, Marcos Treviso, Miguel Vera, and André F. T. Martins. 2019. OpenKiwi: An open source framework for quality estimation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 117–122, Florence, Italy. Association for Computational Linguistics.

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, Melbourne, Australia. Association for Computational Linguistics.

Kyungyul Kim, ByeongMoon Ji, Doyoung Yoon, and Sangheum Hwang. 2021. Self-knowledge distillation with progressive refinement of targets. *Preprint*, arXiv:2006.12000.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations (ICLR)*, San Diega, CA, USA.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Julia Kreutzer, Isaac Caswell, Lisa Wang, Ahsan Wahab, Daan van Esch, Nasanbayar Ulzii-Orshikh, Allahsera Tapo, Nishant Subramani, Artem Sokolov, Claytone Sikasote, Monang Setyawan, Supheakmungkol Sarin, Sokhar Samb, Benoît Sagot, Clara Rivera, Annette Rios, Isabel Papadimitriou, Salomey Osei, Pedro Ortiz Suarez, Iroro Orife, Kelechi Ogueji, Andre Niyongabo Rubungo, Toan Q. Nguyen, Mathias Müller, André Müller, Shamsuddeen Hassan Muhammad, Nanda Muhammad, Ayanda Mnyakeni, Jamshidbek Mirzakhalov, Tapiwanashe Matangira, Colin Leong, Nze Lawson, Sneha Kudugunta, Yacine Jernite, Mathias Jenny, Orhan Firat, Bonaventure F. P. Dossou, Sakhile Dlamini, Nisansa de Silva, Sakine Çabuk Ballı, Stella Biderman, Alessia Battisti, Ahmed Baruwa, Ankur Bapna, Pallavi Baljekar, Israel Abebe Azime, Ayodele Awokoya, Duygu Ataman, Orevaoghene Ahia, Oghenefego Ahia, Sweta Agrawal, and Mofetoluwa Adeyemi. 2022. Quality at a glance: An audit of web-crawled multilingual datasets. *Transactions of the Association for Computational Linguistics*, 10:50–72.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

En-Shiun Lee, Sarubi Thillainathan, Shravan Nayak, Surangika Ranathunga, David Adelani, Ruisi Su, and Arya McCarthy. 2022. Pre-trained multilingual sequence-to-sequence models: A hope for low-resource language translation? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 58–67, Dublin, Ireland. Association for Computational Linguistics.

Tianjian Li, Haoran Xu, Philipp Koehn, Daniel Khashabi, and Kenton Murray. 2024. Error norm truncation: Robust training in the presence of data noise for text generation models. In *The Twelfth International Conference on Learning Representations*.

Yangdi Lu and Wenbo He. 2022. Selc: Self-ensemble label correction improves learning with noisy labels. In *International Joint Conference on Artificial Intelligence*.

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 the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

Jan-Thorsten Peter, David Vilar, Daniel Deutsch, Mara Finkelstein, Juraj Juraska, and Markus Freitag. 2023. There's no data like better data: Using QE metrics for MT data filtering. In *Proceedings of the Eighth Conference on Machine Translation*, pages 561–577, Singapore. Association for Computational Linguistics.

Surangika Ranathunga, Nisansa De Silva, Velayuthan Menan, Aloka Fernando, and Charitha Rathnayake.

2024. Quality does matter: A detailed look at the quality and utility of web-mined parallel corpora. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 860–880, St. Julian's, Malta. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Holger Schwenk and Matthijs Douze. 2017. Learning joint multilingual sentence representations with neural machine translation. In *Rep4NLP@ACL*.

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 *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Wei Wang, Taro Watanabe, Macduff Hughes, Tetsuji Nakagawa, and Ciprian Chelba. 2018. Denoising neural machine translation training with trusted data and online data selection. In *Conference on Machine Translation*.

Xinshao Wang, Yang Hua, Elyor Kodirov, Sankha Subhra Mukherjee, David A. Clifton, and Neil Martin Robertson. 2022. Proselflc: Progressive self label correction towards a low-temperature entropy state. *bioRxiv*.

Haoran Xu, Amr Sharaf, Yunmo Chen, Weiting Tan, Lingfeng Shen, Benjamin Van Durme, Kenton Murray, and Young Jin Kim. 2024. Contrastive preference optimization: Pushing the boundaries of llm performance in machine translation. *Preprint*, arXiv:2401.08417.

# A Data Filters

For LASER (Artetxe and Schwenk, 2018), data filtering scores sentence pairs based on cross-lingual sentence embeddings. To calculate the LASER score for each sentence pair, we generate cross-lingual sentence embeddings using the pre-trained LASER model[7]. The underlying system is trained as a multilingual translation system with a multi-layer bidirectional LSTM encoder and an LSTM decoder without information about the input language on the encoder. The output vectors of the encoder are compressed into a single embedding of fixed length using max-pooling, which is the cross-lingual sentence embedding resulting from the LASER model. The assumption is that two sentences with the same meaning but from different languages will be mapped onto the same embedding vectors. We calculate the LASER score followed by (Chaudhary et al., 2019). The higher the LASER score, the more semantically similar the source and target sentence are.

COMET is a neural framework for training machine translation evaluation models that can function as metrics (Rei et al., 2020). Their framework uses cross-lingual pre-trained language modeling that exploits information from both the source input and the target reference to predict the target translation quality. We use the `reference-free` `wmt-20-qe-da` COMET model as the data filter to score each sentence pair in the training corpus.

Bi-Cleaner is a tool in Python that aims at detecting noisy sentence pairs in a parallel corpus. It indicates the likelihood of a pair of sentences being mutual translations. Sentence pairs considered high-quality are scored near 1, and those considered noisy are scored with 0. We use the multilingual model `bitextor/bicleaner-ai-full-en-xx` from HuggingFace[8] for the pre-filter for all language tasks.

XLM-R is a transformer-based multilingual masked language model pre-trained on text in 100 languages. We extract the sentence embeddings from the source and target with the model from Conneau et al. (2019) and calculate their cosine similarity score as the XLM-R score.

[7]`https://github.com/facebookresearch/LASER/blob/main/nllb/README.md`

[8]`https://huggingface.co/bitextor/bicleaner-ai-full-en-xx`

# B Controlled Generated Misaligned Noise

## B.1 Algorithm

Algorithm 1 generates misaligned noise, controlled by two steps: (1) surface-level features control by word overlap and sentence length; (2) quality control by LASER or COMET.

To save computational resources for calculating the LASER/COMET score for a source sentence with a chunk of target sentences, we first perform surface-level feature control (word overlap and length mismatch) to select a subset of misaligned target candidates. Word overlap is used as a filter to ensure that the misaligned targets share certain surface-level features with the true reference. The same holds for length mismatch.

To avoid overusing the selected misaligned target, we remove the selected target from the chunk of target sentences $T$. In our adequacy evaluation (shown in Appendix B.3 and Table 7), we also show that our misaligned sentences contain only partial meanings of the source sentences. This ensures a low likelihood that the selected misaligned target is a reasonable source sentence translation.

## B.2 Misaligned Noise Samples

Table 6 shows the simulated misaligned samples of Misaligned-LASER and Misaligned-COMET. Overall, the simulated misaligned noise controlled by external models all share certain amounts of semantic meanings compared with the true reference.

| | |
|---|---|
| **SRC** | der Rat kam überein, dass die Kommission die Anwendung dieser Verordnung mit dem Ziel überwacht, etwaige Probleme möglichst schnell festzustellen und zu regeln. |
| **REF** | the Council agreed that the Commission will keep under review the implementation of this Regulation with a view to detecting and addressing any difficulties as soon as possible. |
| **Mis-LASER** | the Commission has therefore acted wisely in exploring every possible avenue to guard against any difficulties and to prepare for any eventualities. |
| **SRC** | Brüssel , 17 März 2015 |
| **REF** | Brussels , 17 March 2015 |
| **Mis-LASER** | Brussels , 4 May 2011 |
| **SRC** | wann möchten Sie im Aeolos Hotel übernachten? |
| **REF** | when would you like to stay at the Aeolos Hotel? |
| **Mis-LASER** | when would you like to stay at the Leenane Hotel? |
| **SRC** | buchen Sie Ihre Unterkunft in Edinburgh today! |
| **REF** | book your accommodation in Edinburgh today! |
| **Mis-COMET** | book your accommodation in Amsterdam today! |
| **SRC** | wir akzeptieren folgende Kreditkarten:Visa, Maestro, Master Card, American Express, JBC, Dinners Club. |
| **REF** | We accept the following credit cards: Visa, Maestro, Master Card, American Express, JBC, Dinners Club. |
| **Mis-COMET** | we accept payments by credit card (Visa, MasterCard, Diners Club), Paypal or transfer. |
| **SRC** | Puchacz Puchacz Spa befindet sich in Niechorze , in einer schönen und malerischen Umgebung , ist lediglich 150m vom Meer entfernt und liegt in der Nähe des Liwia Łuża Sees . |
| **REF** | Puchacz Puchacz Spa is located in Niechorze, in a beautiful and picturesque setting, only 150m from the sea and close to Lake Liwia Łuża. |
| **Mis-COMET** | the Country Hotel Sa Talaia, surrounded by beautiful gardens is located close to San Antonio city and not far away from the historic city of Ibiza |

Table 6: Simulated Misaligned Sentences Samples

## B.3 Adequacy Evaluation

To evaluate the adequacy of the real world and our simulated misalignment noise, we design an annotation guide (see Table 7) to select the overlap meanings between a source sentence with the misaligned target. The simulated misaligned sentence pairs are constructed from the clean corpus WMT2017 De→En, and the real-world misaligned sentences are selected from web-mined Paracrawl datasets. The annotations were conducted by the two PhD students, who are also the authors of this paper, as volunteers without compensation.

## B.4 Token-level Annotation

We conducted a token-level annotation on 50 misaligned and clean sentences, resulting in 480 misaligned tokens and 1557 clean tokens. The annotators must label each token as "clean" or "noisy" given a source and a target sentence. The annotated misaligned and clean sentences are sampled from the sentences used in Section 3.3. The annotations were conducted by the two PhD students, who are also the authors of this paper, as volunteers without compensation.

Table 8 shows that the average *el2n* values for the misaligned tokens are higher than those for clean tokens, in both misaligned and clean samples, further confirming the effectiveness of leveraging the model's self-knowledge to distinguish data noise. Moreover, we also find some clean tokens in clean target sentences do have higher *el2n* values (shown in Table 9). We find that clean tokens with higher *el2n* values tend to be difficult words for the model to learn, e.g., "communication" and "developments".

**Algorithm 1** Misaligned Noise Generation

**Input:** A chunk of parallel and de-duplicate clean data $D$ with N sentence pairs, source and target $(S, T)$; A threshold $k$ for selecting misaligned candidates; A quality controlled model $M \in \{\text{LASER}, \text{COMET}\}$

**Output:** Misaligned data $\bar{D}$ with N sentence pairs source and misaligned target $(S, \bar{T})$.

**for** each source sentence $s_i$ in $S$ **do**
    **Step 1: Surface-level Features Control**
    Initialize a list $L$ of misaligned candidates for $s_i$
    **for** each target sentence $t_{j(j \neq i)}$ in $T$ **do**
        **if** $\text{len}(L) < k$ **then**
            **if** $|\,\text{len}(t_j) - \text{len}(s_i)| < 3$ **and** word overlap ratio$(t_j, t_i) > 0.4$ **then**
                Append $t_j$ to list $L$
            **end if**
        **end if**
    **end for**
    **Step 2: Quality Control**
    Initialize a quality score list $Q$
    **for** each candidate $t_n$ in $L$ **do**
        $\text{score}(s_i, t_n) = M(s_i, t_n)$
        Append score to list $Q$
    **end for**
    Select $t_k$ from $L$ with the highest score in $Q$
    Append the pair $(s_i, t_k)$ to the misaligned data $\bar{D}$
    Remove $t_k$ from targets $T$ to avoid $t_k$ over-reused
**end for**

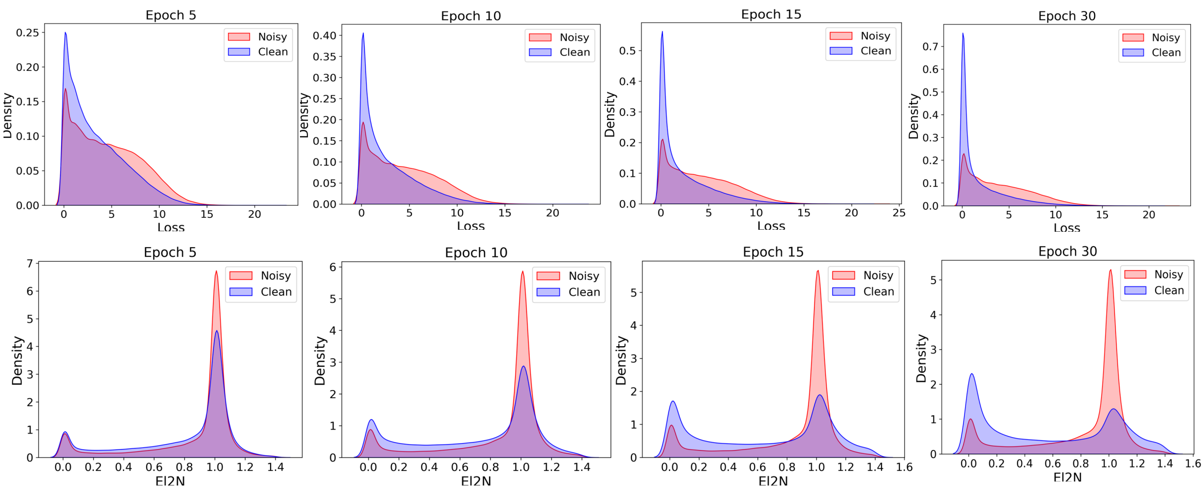


Figure 4: *loss* (above) and *el2n* (below) distribution for clean and misaligned-*LASER* noise samples during the training process (Epoch = 5, 10, 15, 30). Red distribution represents misaligned-*LASER* noise and blue distribution represents the clean data.

| **Questionnaire** |
|---|
| Whether this target translation conveys the same meanings as the source sentence? |
| ○ all meanings ○ most meanings ○ much meanings ○ little meanings ○ no meanings |

Table 7: Questionnaire for human evaluation, where ○ indicate single-item selection. From all meanings to no meanings, the adequacy score scales from 5–1.

| **Misaligned** | **Clean-M** | **Clean-C** |
|---|---|---|
| 1.13 | 0.32 | 0.37 |

Table 8: Average *el2n* values for annotated misaligned and clean tokens. Clean-M: Clean tokens from misaligned samples; Clean-C: Clean tokens from clean samples.

| | |
|---|---|
| **SRC** | _ganz _entschieden _möchte _ich _mich _gegen _den _Ansatz _der _Kommission _wenden _, _wie _er _in _ihrer _Mitteilung _zum _Ausdruck _kommt _. |
| **TGT** | _I _should _also _like _to _firmly _contest _the _Commission _& apos ; s _approach _as _presented _in _its _communication _. |
| **High *el2n*** | _contest, _communication |
| **SRC** | _wir _werden _daher _diesen _Bericht _unterstützen _und _das _Thema _auch _weiterhin _mit _großer _Aufmerksamkeit _verfolgen _. |
| **TGT** | _we _therefore _support _this _report _and _will _continue _to _closely _monitor _developments _. |
| **High *el2n*** | _closely, _monitor, _developments |
| **SRC** | _folglich _muß _bis _zur _Revision _ein _ausreichen der _Zeitraum _ver gehen _. |
| **TGT** | _we _must _therefore _provide _for _a _review _after _a _sufficient _period _. |
| **High *el2n*** | _therefore |

Table 9: Clean sentence that contain tokens with high *el2n* values. Here high *el2n* represents the clean tokens have an *el2n* value exceeding 1.35.

# C Self-Correction Method Design

## C.1 Label Correction in Computer Vision

Our self-correction method is in line with the label correction method in Computer Vision (Wang et al., 2022; Lu and He, 2022). Both approaches are motivated by the idea of correcting data noise using a model's self-knowledge. However, we are the first work to apply this approach specifically in the text de-noise field.

While other work (Kim et al., 2021) highlights another benefit of using the model's predictions to refine the target, i.e. regularization. However, we do not discuss this aspect in our paper. This is because we share different motivations. Our work primarily aims to improve the robustness of training to address the low-quality training data issues instead of regularizing the model.

## C.2 Hyper-Parameter Selection

In Time$(t)$, $\alpha$ decides the inflection point, and $\beta$ adjusts the exponentiation's base and growth speed. Therefore, we fixed $\alpha = -0.6$ and conducted prior experiments to select $\beta$. Table 10 provides the results of different $\beta$ under 30% misaligned noise rations for high-resource and low-resource tasks. We select $\alpha = -0.6$ and $\beta = -6$ for our experiments.

| $\beta$ | High-resource | Low-resource |
|---|---|---|
| -4 | 31.9 | 19.7 |
| -5 | 32.0 | 20.0 |
| **-6** | **32.3** | **20.3** |
| -7 | 31.8 | 20.2 |
| -8 | 31.4 | 20.0 |

Table 10: Hyper-parameter Selection for $\beta$. We report the BLEU scores for different $\beta$ on high-resource task: De→ En and low-resource task: En→Si.

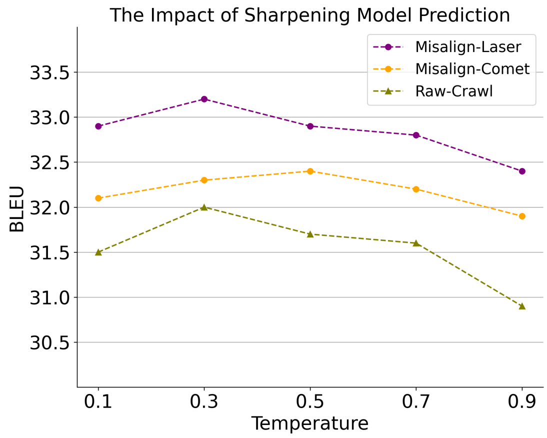


Figure 5: BLEU scores from the self-correction models on De→En task with 30% different types of injected noise with varying $\tau$.

## C.3 The Impact of Sharpening Model Prediction.

Here, we aim to analyze the impact of sharpening model prediction distribution, i.e., different fixed values of $\tau$, to correct the ground truth on translation performance. We train the self-correction models on De→En task with 30% of different types of noise, with varying values of softmax temperature $\tau$. From figure 5, we show that using sharpening model prediction distribution with a smaller $\tau$ achieves better translation performance for all noisy settings. However, the optimal $\tau$ varies when training with different types of noise and thus increases the difficulty of selecting a fixed $\tau$ for different scenarios. This motivates us to design a dynamic $\tau$, which varies automatically in a low range of entropy state over training time. The overall performance in both Section 5.2 and Section 5.3 by using a dynamic $\tau$ also shows its general applicability for different noise scenarios.

# D Training Details.

## D.1 Training and Evaluation

We follow the setup of the Transformer-base and Transformer-big models (Bengio et al., 2015). For each model, the number of layers in the encoder and in the decoder is $N = 6$. We employ $h = 8$ parallel attention layers and heads for the Transformer-base. The dimensionality of input and output is $d_{\text{model}} = 512$, and the inner layer of feed-forward networks has dimensionality $d_{\text{ff}} = 2048$. We employ $h = 16$ parallel attention layers and heads for Transformer-big. The dimensionality of input and output is $d_{\text{model}} = 1024$, and the inner layer of feed-forward networks has dimensionality $d_{\text{ff}} = 4096$.

All models are trained with the Adam optimizer (Kingma and Ba, 2015) for up to 500K steps for high-resource tasks and 100K steps for low-resource tasks, with a learning rate of 5e-4 and an inverse square root scheduler. A dropout rate of 0.3 and label smoothing of 0.2 are used. Each model is trained on one NVIDIA A6000 GPU with a batch size of 25K tokens. We choose the best checkpoint according to the average validation loss of all language pairs. The data is tokenized with the SentencePiece tool (Kudo and Richardson, 2018), and we build a shared vocabulary of 32K tokens. For evaluation, we employ beam search decoding with a beam size of 5. BLEU scores are computed using detokenized case-sensitive SacreBLEU[9].

## D.2 Dataset Details

### D.2.1 Simulated Noise Setting

Table 11 shows the training and evaluation dataset details for clean training corpus in simulated noisy experiments in Section 5.2.

| Translation Task | Training Source | Dev Set | Test Set |
|---|---|---|---|
| De→En | WMT2017 (5.8M) | NewsTest2016 | NewsTest2017 |
| En→Si | OPUS (0.9M) | OPUS | OPUS |

Table 11: The clean training corpus and evaluation dataset details for experiments in Section 5.2.

### D.2.2 Real-World Noise Setting

For Paracrawl, the language pairs are: en→fr (French), en→si (Sinhala), en→sw (Swahili), and en→km (Khmer). For CCAligned, the language pairs are en→tr (Turkish), en→es (Spanish), and en→be (Belarusian). For the high-resource language pairs: en→fr, en→tr, en→es, we randomly sample 5M sentence pairs as the training corpus. For medium and low-resource language pairs, we use the original corpus size.

# E Chrf++ and COMET Scores

Table 12, 13, and 14 shows the COMET (`Unbabel/wmt22-comet-da`) and Chrf++ scores for all experiments.

[9] nrefs:1|case:mixed|eff:no|tok:13a|smooth:exp|version:2.3.1

| | | COMET | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Misaligned-LASER** | | | **Misaligned-COMET** | | | **Raw-Crawl Data** | | |
| | | 10% | 30% | 50% | 10% | 30% | 50% | 10% | 30% | 50% |
| **Baseline** | *with* noise | 77.8$^*$ | 77.0$^*$ | 76.1$^*$ | 77.6$^*$ | 76.5$^*$ | 75.5$^*$ | 77.9$^*$ | 77.1$^*$ | 75.8$^*$ |
| **Oracle** | *w/o* noise | 79.5 | 79.0 | 78.6 | 79.5 | 79.0 | 78.6 | 79.5 | 79.0 | 78.6 |
| **Pre-Filter** | LASER | 78.0$^*$ | 76.9$^*$ | 75.6$^*$ | 78.2$^*$ | **78.0** | 76.0$^*$ | 78.0$^*$ | 77.8$^*$ | 76.9 |
| | COMET | 77.9$^*$ | 77.5$^*$ | 76.3$^*$ | 77.5$^*$ | 76.3$^*$ | 74.0$^*$ | 78.0$^*$ | 76.8$^*$ | 75.6$^*$ |
| **Truncation** | *loss* | 78.3$^*$ | 76.5$^*$ | 76.2$^*$ | 78.0$^*$ | 76.3$^*$ | 75.0$^*$ | 78.0$^*$ | 77.2$^*$ | 76.6$^*$ |
| | *el2n* | 78.3$^*$ | 78.3 | 76.5$^*$ | 78.1$^*$ | 76.1$^*$ | 76.0$^*$ | 78.2$^*$ | 77.5$^*$ | 76.2$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | 79.0 | 78.5 | 76.8 | 78.5 | 77.6 | 76.2 | 78.8 | 78.1 | 76.5 |
| | dynamic $\tau$ | **79.1** | **78.6** | **77.0** | **78.7** | 77.7 | **76.6** | **79.0** | **78.3** | **77.0** |
| | | **Chrf++** | | | | | | | | |
| **Baseline** | *with* noise | 55.5$^*$ | 54.9$^*$ | 54.1$^*$ | 55.1$^*$ | 54.7$^*$ | 52.5$^*$ | 55.0$^*$ | 54.9$^*$ | 53.6$^*$ |
| **Oracle** | *w/o* noise | 57.2 | 56.9 | 55.5 | 57.2 | 56.9 | 55.5 | 57.2 | 56.9 | 55.5 |
| **Pre-Filter** | LASER | 56.5$^*$ | 54.5$^*$ | 53.4$^*$ | 56.3 | **56.0** | 52.6$^*$ | 55.2$^*$ | 55.0$^*$ | 54.3$^*$ |
| | COMET | 56.0$^*$ | 54.2$^*$ | 53.0$^*$ | 55.0$^*$ | 54.2$^*$ | 51.9$^*$ | 55.2$^*$ | 54.2$^*$ | 52.8$^*$ |
| **Truncation** | *loss* | 56.0$^*$ | 54.3$^*$ | 54.2$^*$ | 55.5$^*$ | 54.1$^*$ | 52.0$^*$ | 55.5$^*$ | 55.0$^*$ | 54.0$^*$ |
| | *el2n* | 56.1$^*$ | 55.2$^*$ | 54.2$^*$ | 55.5$^*$ | 55.0$^*$ | 52.0$^*$ | 56.2$^*$ | 55.0$^*$ | 54.2$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | 56.8 | **56.5** | 54.3 | **56.6** | 55.2 | 52.8 | 56.6 | 55.5 | 54.0 |
| | dynamic $\tau$ | **56.9** | 56.2 | **54.6** | 56.4 | 55.6 | **53.0** | **56.7** | **55.8** | **54.9** |

Table 12: COMET and Chrf++ scores of high-resource De $\rightarrow$ En translation task with different types of noise. The COMET score of full clean training corpus (5.8M) De $\rightarrow$ En is 80.0. The Chrf++ score of full clean training corpus (5.8M) De $\rightarrow$ En is 57.2. $*$ signifies that our self-correction method is significantly better (p-value < 0.05) than the baseline.

| | | COMET | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | **Misaligned-LASER** | | | **Misaligned-COMET** | | | **Raw-Crawl Data** | | |
| | | 10% | 30% | 50% | 10% | 30% | 50% | 10% | 30% | 50% |
| **Baseline** | *with* noise | 79.8$^*$ | 79.0 | 77.8$^*$ | 79.7 | 75.9$^*$ | 71.6$^*$ | 79.7$^*$ | 79.5$^*$ | 78.3$^*$ |
| **Oracle** | *w/o* noise | 79.8 | 79.4 | 78.9 | 79.8 | 79.4 | 78.9 | 79.8 | 79.4 | 78.9 |
| **Pre-Filter** | LASER | 79.5$^*$ | 78.5$^*$ | 77.0$^*$ | 79.5$^*$ | 76.2$^*$ | **74.7** | 79.8$^*$ | 79.8$^*$ | 79.0 |
| | COMET | 79.6$^*$ | 78.8$^*$ | 76.8$^*$ | 79.2$^*$ | 76.0$^*$ | 71.0$^*$ | 79.5$^*$ | 79.0$^*$ | 77.8$^*$ |
| **Truncation** | *loss* | 79.9 | 78.4$^*$ | 78.0$^*$ | 79.0$^*$ | 75.6$^*$ | 71.2$^*$ | 79.8$^*$ | 79.4$^*$ | 78.6$^*$ |
| | *el2n* | 80.1 | 79.1 | 78.2$^*$ | 79.8 | 76.2$^*$ | 72.3$^*$ | 79.9 | 79.5$^*$ | 78.8$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | **80.3** | 79.0 | 78.5 | 79.9 | 77.0 | 74.0 | **80.3** | 79.8 | **79.5** |
| | dynamic $\tau$ | 80.1 | **79.2** | **78.8** | **79.9** | **77.1** | 74.6 | 80.2 | **80.1** | 79.2 |
| | | **Chrf++** | | | | | | | | |
| **Baseline** | *with* noise | 35.7 | 34.0$^*$ | 33.0$^*$ | 34.9$^*$ | 30.1$^*$ | 24.2$^*$ | 35.6$^*$ | 34.0$^*$ | 32.7$^*$ |
| **Oracle** | *w/o* noise | 35.9 | 34.6 | 34.2 | 35.9 | 34.6 | 34.2 | 35.9 | 34.6 | 34.2 |
| **Pre-Filter** | LASER | 35.4$^*$ | 33.2$^*$ | 32.5$^*$ | 35.4$^*$ | 31.2 | 28.0 | 35.7 | 34.2$^*$ | 33.0$^*$ |
| | COMET | 35.4$^*$ | 33.5$^*$ | 32.6$^*$ | 33.6$^*$ | 29.5$^*$ | 23.8$^*$ | 35.4$^*$ | 33.8$^*$ | 32.5$^*$ |
| **Truncation** | *loss* | 35.8 | 33.6$^*$ | 33.2$^*$ | 35.3$^*$ | 30.2$^*$ | 25.8$^*$ | 35.7 | 34.2$^*$ | 32.8$^*$ |
| | *el2n* | 35.6 | 34.1$^*$ | 33.3$^*$ | 35.6 | 30.4$^*$ | 26.0$^*$ | 35.6 | 34.1$^*$ | 32.8$^*$ |
| **Self-Correction (Ours)** | fixed $\tau = 0.5$ | **36.0** | 34.3 | 33.3 | 35.5 | 31.0 | 27.0 | **36.0** | 34.8 | **33.8** |
| | dynamic $\tau$ | 35.8 | **34.4** | **33.6** | **35.8** | **31.5** | **28.3** | 35.8 | **35.0** | 33.4 |

Table 13: COMET and Chrf++ scores of low-resource En $\rightarrow$ Si translation task with different types of noise. The COMET score of full clean training corpus (0.9M) En $\rightarrow$ Si is 82.0. The Chrf++ score of full clean training corpus (0.9M) En $\rightarrow$ Si is 37.0. $*$ signifies that our self-correction method is significantly better (p-value < 0.05) than the baseline.

| | | COMET | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | en→fr♡ | en→tr† | en→es† | en→be† | en→si♡ | en→sw♡ | en→km♡ | Avg. |
| **Misaligned Rate (%)** | | 10% | 44% | 22% | 10% | 62% | 11% | 18% | - |
| **Corpus Size (M)** | | 5M | 5M | 5M | 1.1M | 210K | 130K | 60K | - |
| **Baseline** | | 80.0* | 82.0* | 76.5* | 68.3* | 59.6* | 59.0* | 73.6* | 71.3 |
| **Pre-Filter** | LASER | 81.0* | 81.3* | <u>76.7</u>* | 67.4* | 59.7* | 58.3* | 73.6* | 71.1 |
| | COMET | 80.5* | 81.0* | 76.0* | <u>68.5</u>* | 59.5* | 58.1* | 73.2* | 71.0 |
| **Truncation** | *loss* | 81.0* | 82.2* | 76.8* | 67.6* | 59.0* | 58.8* | 73.0* | 71.2 |
| | *el2n* | 80.2* | 82.1* | 76.2* | 68.6* | 60.0* | 58.6* | 72.8* | 71.2 |
| **Self-Correction** | fixed $\tau = 0.5$ | <u>81.2</u> | <u>82.5</u> | 76.4 | 68.4 | <u>63.0</u> | <u>61.0</u> | <u>74.5</u> | <u>72.4</u> |
| | dynamic $\tau$ | **81.6** | **83.0** | **77.9** | **68.9** | **63.6** | **61.4** | **75.0** | **73.0** |
| | | ChrF++ | | | | | | | |
| **Baseline** | | 67.3* | 54.8* | 49.1* | 36.5* | 20.2* | 37.9* | 15.6* | 40.2 |
| **Pre-Filter** | LASER | 67.9* | 54.6* | <u>49.6</u>* | 36.1* | 21.7* | 37.5* | 14.7* | 40.3 |
| | COMET | 67.6* | 54.3* | 49.6* | 36.2* | 20.6* | 37.2* | 15.0* | 40.1 |
| **Truncation** | *loss* | 67.4* | 55.2 | 49.2* | 36.3* | 20.0* | 37.9* | 13.4* | 39.9 |
| | *el2n* | 67.6* | <u>55.2</u> | 49.5* | 36.5* | 20.6* | 37.3* | 13.0* | 40.1 |
| **Self-Correction** | fixed $\tau = 0.5$ | <u>68.0</u> | 54.9 | 49.6 | <u>36.8</u> | **24.0** | <u>41.0</u> | <u>16.5</u> | <u>41.5</u> |
| | dynamic $\tau$ | **68.2** | **55.4** | **50.0** | **37.2** | <u>22.2</u> | **42.3** | **16.8** | **41.7** |

Table 14: COMET and Chrf++ scores on real-world web-mined corpora. For pre-filter methods, we remove 20% of the training samples with the lowest scores. † denotes language pairs from CCAligned V1.0. ♡denotes language pairs from ParaCrawl V7.1. The misaligned noise rate for different language pairs is reported from Kreutzer et al. (2022). $*$ signifies that our self-correction method is significantly better (p-value < 0.05) than the baseline.