Separating Grains from the Chaff: Using Data Filtering to Improve Multilingual Translation for Low-Resourced African Languages

Idris Abdulmumin1,2, Michael Beukman3, Jesujoba O. Alabi4, Chris Emezue5,6, Everlyn Asiko7,8, Tosin Adewumi9, Shamsudddeen Hassan Muhammad10,11, Mofetoluwa Adeyemi, Oreen Yousuf11, Sahib Singh12, Tajuddeen Rabiu Gwadabe2,13

All: MasakhaneNLP, 1Ahmadu Bello University, Zaria, Nigeria, 2HausaNLP, 3University of the Witwatersrand, South Africa, 4Saarland University, Germany, 5TUM, Germany, 6Mila - Quebec AI Institute, 7University of Cape Town, South Africa, 8African Institute for Mathematical Sciences, 9Luleå University of Technology, Sweden, 10LIAAD-INESC TEC, Porto, Portugal, 11Uppsala University, Sweden, 12Ford Motor Company, 13University of Chinese Academy of Sciences, China

iabdulmumin@abu.edu.ng

Abstract

We participated in the WMT 2022 Large-Scale Machine Translation Evaluation for the African Languages Shared Task. This work describes our approach, which is based on filtering the given noisy data using a sentence-pair classifier that was built by fine-tuning a pre-trained language model. To train the classifier, we obtain positive samples (i.e. high-quality parallel sentences) from a gold-standard curated dataset and extract negative samples (i.e. low-quality parallel sentences) from automatically aligned parallel data by choosing sentences with low alignment scores. Our final machine translation model was then trained on filtered data, instead of the entire noisy dataset. We empirically validate our approach by evaluating on two common datasets and show that data filtering generally improves overall translation quality, in some cases even significantly.

1 Introduction

This paper presents Masakhane NLP’s submission to the WMT 2022 large-scale machine translation evaluation for African languages. We participated in the constrained translation task and chose to focus on a subset of all the language pairs considered for this task due to resource constraints. We specifically explore the language directions {hau, ibo, lug, swa, tsn, yor, zul}↔eng and wol↔fra. and submitted our primary and secondary systems which were competitive with other submissions for this task.

Machine translation has received much attention recently, especially for low-resourced languages (Adelani et al., 2022a; Fan et al., 2021; Haddow et al., 2022; Hoang et al., 2018; Nekoto et al., 2020). A promising approach for such setups is to fine-tune large pre-trained language models on the available small amount of translation data (Neubig and Hu, 2018; Adelani et al., 2021a, 2022a). While most of these language models are trained on predominantly high-resourced language datasets (Conneau et al., 2020; Devlin et al., 2019; Radford et al., 2018), there have been a few models that were pre-trained (Ogujje et al., 2021) or adaptively fine-tuned (Alabi et al., 2022) only on low-resourced languages.

Recent works have tried, successfully, to supplement the existing small amounts of natural data in low-resource languages with artificially generated parallel data. For instance, in machine translation, Sennrich et al. (2016) and Ueffing (2006) padded the true parallel data with automatic translations of monolingual sentences through back-translation and self-learning respectively. Others, such as Bañón et al. (2020); El-Kishky et al. (2020); and Schw enk et al. (2021), have used different approaches for detecting potentially aligned sentences within web datasets. While significant improvements have been achieved with these synthetic datasets, an in-depth investigation by Kreutzer et al. 2022 has found them to be fraught with many issues, such as misalignment, wrongful language codes, etc.

Similarly, research has shown that data quality plays an important role in the performance of natural language processing (NLP) models, in machine translation specifically  (Arora et al., 2021; Dutta et al., 2020; Hasan et al., 2020; Tchistia kova et al., 2021), but also more generally in other NLP tasks (Abdul-Rauf et al., 2012; Alabi et al., 2020). It was found that often times, models that were trained on smaller amounts of high-quality data outperform their counterparts that are trained on larger amounts of noisy datasets (Gascó et al., 2012; Przystupa and Abdul-Mageed, 2019; Abdulmunim et al., 2022; de Gibert et al., 2022). This has led to many studies (Et et al., 2015) and prior WMT tasks (Koehn et al., 2018, 2019, 2020) that...
attempt to find ways to improve the quality of existing data, which, as mentioned before, is often rife with errors.

Therefore, in our submission to the shared task, we experimented with filtering web-mined data for African languages using pre-trained language models and evaluated the effect of using this filtered data on machine translation performance. We defined our filtering approach as a sentence-pair binary classification task and fine-tuned a pre-trained language model using positive and negative samples. We used sentences from the high-quality MAFAND-MT (Adelani et al., 2022a) dataset (which was included in the training data for the constrained task) as positive examples and created negative examples by extracting sentences with low language-agnostic sentence representations (LASER) (Artetxe and Schwenk, 2019b) alignment scores from the wmt22-african (NLLB Team et al., 2022) corpus that was provided for this task. Our results highlight the importance of filtering on the quality of the final machine translation system. We also detail how to create a high-quality filter for African languages using a few gold-standard parallel sentences. We release our codes on GitHub.1

The rest of the paper is organized as follows: in Section 2, we review related work, and in Section 3, we present the dataset we used. Section 4 provides an overview of the bitext filtering approach, while Section 5 details experimental settings and the translation model architecture. In Section 6, we evaluate the model’s performance, and lastly in Section 7, we conclude and highlight some future research directions.

2 Related Work

One of the difficulties when dealing with low-resource settings, as we do here, is that high-quality parallel texts are particularly scarce (Koehn and Knowles, 2017). To curate data for such language pairs, methods for automatically mining parallel text from the web using heuristics (Resnik, 1999) or latent space and similarity-based filters (Artetxe and Schwenk, 2019a; Schwenk et al., 2021) have been proposed. These have led to the curation of publicly available web-mined datasets such as CCAigned (El-Kishky et al., 2020), CC-Matrix (Fan et al., 2021; Schwenk et al., 2021), ParaCrawl (Esplà et al., 2019), and WikiMatrix (Schwenk et al., 2019) to mention just a few.

However, the recent research work by Kreutzer et al. (2022) shows that the automatically aligned and mined parallel bitexts, especially for low-resource language pairs, contain various degrees of errors and less than half of the data are of good quality. Additionally, many approaches generate large amounts of synthetic data, often through back-translation, where synthetic parallel data is generated by automatically translating monolingual data (Bojar and Tammyn, 2011; Lambert et al., 2011; Sennrich et al., 2016). While additional data has the potential to improve the trained models, these synthetic datasets are often of low quality (Xu et al., 2019). These observations have led to an increased interest in the automatic filtering of noisy bitexts as a key research topic in machine translation (MT).

One approach to improve data quality is to filter out the noisy or invalid parts of a large corpus, keeping only a high-quality subset thereof (Abdulmumin et al., 2021). In this vein, numerous filtering methods have been developed (Axelrod et al., 2011; Eetemadi and Toutanova, 2015; Junczys-Dowmunt, 2018). For instance, Xu et al. (2019) use the cosine similarity between sentence embeddings as a measure of how closely aligned two sentences are. Imankulova et al. (2017) perform back-translation and then filter based on the sentence-level BLEU score, keeping only those sentences with a high BLEU. Similarly, Adjeisah et al. (2021) perform a round-trip translation and only use the sentence pair if it is sufficiently close to the original sentence, according to a chosen similarity measure. There has also been work on alignment between two parallel corpora, and Hasan et al. (2020) uses the LASER score2 to evaluate alignment, and filter out all sentences below a specific threshold.

3 Datasets

We participated in the constrained translation track and used only the provided dataset. We present the various dataset used, their sizes and corresponding sources in Table 9 in Appendix A. For our experiment, we selected 8 language pairs and developed different multilingual machine translation systems for them. These language pairs are {hau, ibo, lug, swa, tsn, yor, zul}↔eng and wol↔fra. According to the recommendation

---

1https://github.com/abumafrim/WMT22-MasaKhan

2https://github.com/facebookresearch/LASER
Table 1: Examples of noise in the auto-aligned bitext

<table>
<thead>
<tr>
<th>Direction</th>
<th>Parallel sentences</th>
<th>Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>eng → hau</td>
<td>src: I booked the house for my husband’s family as we were getting married in Ericeira. tgt: nsarr da aba a a ka kasarr ni ila ure imbarr yin ngbangbangu.</td>
<td>tgt is not a Hausa sentence</td>
</tr>
<tr>
<td>eng → hau</td>
<td>src: “Go hunt, and may the light be with you.”** tgt: “Zo, zo muje, ke kika hada fitinar ke za ki wareware ta.”**</td>
<td>tgt is not a translation of the src</td>
</tr>
<tr>
<td>eng → hau</td>
<td>src: The Moslem creed. tgt: Musa Aminta</td>
<td>mismatched named entities</td>
</tr>
<tr>
<td>eng → hau</td>
<td>src: Israel tgt: 00000000000000000000000000000000</td>
<td>mistranslation; foreign characters</td>
</tr>
</tbody>
</table>

Table 2: Training data after filtering using heuristics et al., 2021) and others highlighted in Table 9.

<table>
<thead>
<tr>
<th>Language pair</th>
<th>Data size</th>
<th>% of original</th>
</tr>
</thead>
<tbody>
<tr>
<td>eng hau</td>
<td>9,122,559</td>
<td>99.9</td>
</tr>
<tr>
<td>ibo</td>
<td>520,544</td>
<td>99.6</td>
</tr>
</tbody>
</table>
lug | 3,511,275 | 99.8 |
swa | 32,898,533 | 99.6 |
tsn | 6,036,656 | 99.1 |
yor | 1,718,105 | 99.3 |
zul | 4,142,146 | 97.6 |
|fra wol | 237,348 | 100.0 |

3https://github.com/masakhane-io/lafand-mt.git
4https://github.com/mozilla-l10n/mt-training-data
5https://tanzil.net/tech/
6https://huggingface.co/datasets/allemai/wmt22_african
7https://github.com/pavanpankaj/Web-Crawl-African

3 of Kreutzer et al. (2022), we carefully examined the training dataset provided by manual inspection and divided it into two categories based on the source of the data and the amount of noise included therein. In the following subsections, we describe these two categories of data.

3.1 Clean Bitext

This category of training data comprises all the datasets that are considered to be manually curated. The datasets in this category include: bibleuedin (Christodouloupolous and Steedman, 2015), MAFAND-MT, QED (Abdelali et al., 2014), Mozilla-I10n, and several others listed in Table 9. The clean bitext consists of sentences mostly in the news and religious domains, with a few in the health, education, and technology domains. We also refer to the clean bitext as True Parallel in this paper.

3.2 Noisy Bitext

We categorized all the automatically aligned bitext as noisy bitext. This also includes the LASER filtered data. The sentences in this category make up the majority of the training dataset, making up 99.2% of the total training data. The datasets in this category include: CCAIgined, CCMatrix, LASER wmt22_african, WebCrawl African, and the following datasets from OPUS (Tiedemann, 2012): MultiCCAIgined (El-Kishky et al., 2020), TED2020 (Reimers and Gurevych, 2020), WikiMatrix (Schwenk et al., 2019), XLEnt (El-Kishky et al., 2021) and others highlighted in Table 9.

On manual inspection, however, we found numerous issues with the data, including non-parallel sentences, sentences that consist of only numbers and/or punctuation, sentences in different languages, etc. Examples of noise in the auto-aligned data can be seen in Table 1.

3.3 Validation and Test Data

For the optimization of our translation systems, we combined the FLORES-101 (Goyal et al., 2022) and MAFAND-MT (Adelani et al., 2022a) development sets for each of the 8 language pairs. To compare the performance of the developed MT engines, we evaluated on the FLORES-101 devtest set and the MAFAND-MT test set.

4 Parallel Data Filtering

To attempt to deal with the highly noisy data, we opted to use filtering techniques to remove many invalid or incorrectly aligned sentences, similar to prior work (Arora et al., 2021; Hasan et al., 2020; Xu et al., 2019). We first used some simple heuristic approaches, described in Section 4.1, and then progress to an automatic filtering method, detailed in Section 4.2.

4.1 Heuristics

We filtered sentences that consist of only numbers and/or punctuation marks. After filtering, the statis-
We fine-tuned two pre-trained language models, AfroXLMR, on the other hand, was chosen because ALBERT (Lan et al., 2020) and AfroXLMR (Alabi et al., 2020), which is a gold-standard parallel dataset, as positive examples. The distribution of the train, dev and test sets from the MAFAND-MT dataset, we adopted an automatic approach to determine the quality of parallel sentences to train our translation models. The approach we adopted is sentence-pair binary classification (Nguyen et al., 2021), where we used a transformer-based model to predict the probability that two aligned sentences are actual translations of each other. We explain the process of training data generation and the experimental choices for building the filtering model.

4.2 Automatic Filtering

Due to the large size of the automatically aligned dataset, we adopted an automatic approach to determine the quality of parallel sentences to train our translation models. The approach we adopted is sentence-pair binary classification (Nguyen et al., 2021), where we used a transformer-based model to predict the probability that two aligned sentences are actual translations of each other. We explain the process of training data generation and the experimental choices for building the filtering model.

4.2.1 Positive and negative samples

To create the training and evaluation data for the sentence-pair classification-based filtering, we generated positive and negative samples from the training data available for this task. We used the train, dev and test sets from the MAFAND-MT dataset, which is a gold-standard parallel dataset, as positive examples. For the negative examples, however, we sorted the sentences in the wmt22_african dataset that was provided for this task based on their LASER alignment scores, and selected the least scored sentences in equal amounts to each of the positive examples. The distribution of the train, dev and test samples is presented in Table 3.

4.2.2 Model

We fine-tuned two pre-trained language models, ALBERT (Lan et al., 2020) and AfroXLMR (Alabi et al., 2022) for the sentence pair binary classification task. ALBERT was selected based on its performance on downstream NLP tasks (Lan et al., 2020), even though it has fewer parameters than other BERT-based models (Nguyen et al., 2021). AfroXLMR, on the other hand, was chosen because it was trained on African languages (Alabi et al., 2022), and such a setup has been shown to improve performance on downstream tasks for these languages (Adelani et al., 2022a).

### Table 3: Sentence-pair classification training data: a mixture of MAFAND-MT sentences, taken as positive samples, and wmt22_african (worst pairs based on LASER scores), taken as negative samples.

<table>
<thead>
<tr>
<th>Data</th>
<th>tsvl</th>
<th>ibol</th>
<th>zulu</th>
<th>tsvl</th>
<th>eng</th>
<th>fra</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train</td>
<td>6,198</td>
<td>13,998</td>
<td>8,152</td>
<td>61,596</td>
<td>4,202</td>
<td>13,290</td>
</tr>
<tr>
<td>Dev</td>
<td>2,602</td>
<td>3,002</td>
<td>3,002</td>
<td>3,604</td>
<td>2,696</td>
<td>3,000</td>
</tr>
<tr>
<td>Test</td>
<td>3,002</td>
<td>3,002</td>
<td>3,002</td>
<td>3,672</td>
<td>3,002</td>
<td>3,118</td>
</tr>
</tbody>
</table>

4.3 Filter Training Setup

The filtering models were trained to accept a pair of sentences from the source and target languages. During training, the [CLS] token hidden representation of the input sentence pairs is fed into a linear Layer and the model is optimized using binary cross entropy loss. However, at inference time, we add a sigmoid layer to the output to predict a number between 0.0 and 1.0 indicating the likelihood of the bitexts being translations of each other. We fine-tuned these models using each language’s train split of positive and negative samples, then evaluated performance on the test set while optimizing on the development set.

The performance of the various automatic filtering models and the subsequent sizes of the filtered datasets for the 8 language pairs are shown in Table 4. This table shows the number of sentence pairs the models classified as actual translation pairs using a threshold of 0.5 and 0.7 as well as the F1 score when using the 0.5 threshold. Additionally, in Table 5, we show the number of sentences that were classified by two or all three of the models as being high-quality.

5 MT Experiments

To evaluate the effect of our filtering techniques, we trained some multilingual NMT models for the 8 language pairs that we have selected for this task. In the following subsections, we highlight the model architectures, training setups, and different multilingual models that were trained.

5.1 Model Architecture

For our experiments, we fine-tune M2M-100 (Fan et al., 2021) on different subsets of the provided data. M2M-100 is a pretrained translation model trained on several languages including African languages, as such it has seen all the languages we have chosen for this task during pre-training. We use the model with 418M parameters.

5.2 Training Setup

We fine-tuned the M2M-100 model based on the implementation within the Fairseq toolkit (Ott et al., 2020).
We train many-to-many (M2M) translation models was only partially cleaned, as evidenced in Table 2, (Kudo and Richardson, 2018) tokenizer. 

5.2.1 Baseline models  

To evaluate our models and to choose the best checkpoints, we used the BLEU score (Papineni et al., 2002) calculated with the SacreBLEU (Post, 2018) implementation. In addition, we also evaluated the models using CHRF (Popović, 2015).

### 5.2.2 Models on filtered data only

To evaluate the effect of the filtered data on the quality of the translation output, we train M2M models on the filtered data from the different models using a threshold of 0.5 and 0.7.

### 5.2.3 Models on filtered and clean data

We went further to train multilingual models on the concatenation of the noisy and clean text, and on the reverse was true. This is likely due to the fact that the MAFAND-MT data is present in the clean bitext performed impressively on the two evaluation datasets, despite the limited dataset size. On average, the baseline model trained on the clean bitext compared to training models only on data.

We trained these baseline models to compare and measure the efficacy of our filtering technique on the quality of the translation models. We submitted the model in (i) as our secondary system for this task.

5.2.3 Models on filtered and clean data

We used batch sizes of 2,048 tokens, a maximum sentence length of 1,024, and a dropout of 0.3. For optimization, we used Adam (Kingma and Ba, 2015) with \( \beta_1 = 0.9 \) and \( \beta_2 = 0.998 \), a learning rate of \( 5e^{-5} \) and a warmup of 2,500 updates. The optimizer uses a label-smoothed cross-entropy loss function with a label-smoothing value of 0.2. All models were trained for a maximum of 1,000,000 update steps. We tokenized all data using the model’s SentencePiece (Kudo and Richardson, 2018) tokenzier.

To evaluate our models and to choose the best checkpoints, we used the BLEU score (Papineni et al., 2002) calculated with the SacreBLEU (Post, 2018) implementation. In addition, we also evaluated the models using CHRF (Popović, 2015).

### 6 Results and Discussion

In Tables 6 and 7, we report the BLEU and CHRF scores obtained by the different models that we trained, as evaluated on the FLORES-101 devtest and MAFAND-MT test datasets, respectively.

### 6.1 Baseline Models

On average, the baseline model trained on the clean bitext performed impressively on the two evaluation datasets, despite the limited dataset size. On MAFAND-MT, the model trained on the clean bitext obtained a higher BLEU score than the model trained on the noisy bitext, and on FLORES-101, the reverse was true. This is likely due to the fact that the MAFAND-MT data is present in the clean
bitext, and that the noisy bitext contains sentences that were taken from the web, including Wikipedia, which is the source of the FLORES-101 dataset. When we compared the model trained on the clean bitext to the model trained on the noisy bitext, we saw between a +1 and +2 improvement on FLORES-101 and between +5 and +8 improvement on MAFAND-MT for lug, wol, and yor. This confirms not only the importance of the data domain, but also the importance of data quality on the quality of the machine translation output.

After mixing the two datasets, the performance improved over using only the clean bitext by more than 6 BLEU on hau↔eng, and almost 3 BLEU on average across all languages on FLORES. The performance, though, was similar to using only the noisy bitext. On the MAFAND-MT test set, however, the performance deteriorated by almost 2 BLEU when compared to training on the clean bitext only. At language-pair level, eng→ibo was affected more (−9.14 BLEU), followed by eng→wol, whereas yor→eng benefited tremendously (+17.83 BLEU). On average, training on the two bitexts marginally improves over using only the noisy bitext, and this is consistent on all the test sets.

Investigating the results in more depth, we found that the BLEU scores of the models are lower when translating into an African language, similar to the findings of Adelani et al. (2022a). This effect is exacerbated for the languages with the fewest parallel sentences, such as lug, wol, and yor, except for ibo, which overall has the second-fewest parallel sentences, as shown in Table 9.

### 6.2 Data Filtering Analysis

We generally see that more filtering results in improved performance, corresponding to removing more noisy sentences from the data. Using less filtering, with a threshold of 0.5, generally performed slightly worse than using a threshold of 0.7. Both of these settings outperformed (a) using no filtering and (b) using no additional data.

We can also see the effect of the filtering steps on the training data in Tables 2 and 4. Filtering the data using heuristics resulted in only a small portion of the data being filtered out. Using the classifier, however, caused a large amount of noisy data to be removed. When looking at the F1 scores of the classification models, we can see that ALBERT-xlarge has the lowest F1, followed by ALBERT-base and AfroXLMR-base. Looking at Table 5, we can see that ALBERT-xlarge is also the most strict filter, removing the most data, whereas AfroXLMR-base removes the least amount of data. Interestingly, the number of sentences marked as high-quality by all three models is surprisingly low, possibly indicating that these different models (particularly ALBERT-xlarge and AfroXLMR-base) focus on different features of the data.

Finally, we saw that a higher threshold resulted in improved translation performance, but ALBERT-xlarge (which is quite strict) had a lower F1 than the

<table>
<thead>
<tr>
<th>Models</th>
<th>hau</th>
<th>ibo</th>
<th>lug</th>
<th>swa</th>
<th>tan</th>
<th>yor</th>
<th>zul</th>
<th>wol</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>eng→x</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clean bitext</td>
<td>9.30</td>
<td>13.19</td>
<td>4.00</td>
<td>23.17</td>
<td>8.56</td>
<td>3.60</td>
<td>9.43</td>
<td>3.56</td>
</tr>
<tr>
<td>Noisy bitext</td>
<td>15.32</td>
<td>10.77</td>
<td>2.14</td>
<td>30.64</td>
<td>12.87</td>
<td>2.57</td>
<td>12.35</td>
<td>0.69</td>
</tr>
<tr>
<td>Clean + Noisy bitext</td>
<td>15.34</td>
<td>11.37</td>
<td>2.40</td>
<td>30.48</td>
<td>13.31</td>
<td>2.48</td>
<td>12.61</td>
<td>0.73</td>
</tr>
<tr>
<td>Filtered only</td>
<td>16.43</td>
<td>15.38</td>
<td>2.54</td>
<td>29.89</td>
<td>16.31</td>
<td>3.00</td>
<td>15.18</td>
<td>0.65</td>
</tr>
<tr>
<td>Filtered + Clean bitext</td>
<td>16.05</td>
<td>15.01</td>
<td>3.22</td>
<td>33.31</td>
<td>15.96</td>
<td>3.08</td>
<td>14.97</td>
<td>1.99</td>
</tr>
<tr>
<td>Filtered only</td>
<td>16.55</td>
<td>15.70</td>
<td>3.45</td>
<td>31.97</td>
<td>16.31</td>
<td>3.16</td>
<td>15.50</td>
<td>2.12</td>
</tr>
<tr>
<td>Filtered + Clean bitext</td>
<td>16.05</td>
<td>15.01</td>
<td>3.22</td>
<td>33.31</td>
<td>15.96</td>
<td>3.08</td>
<td>14.97</td>
<td>1.99</td>
</tr>
<tr>
<td>Filtered + Clean bitext</td>
<td>16.55</td>
<td>15.70</td>
<td>3.45</td>
<td>31.97</td>
<td>16.31</td>
<td>3.16</td>
<td>15.50</td>
<td>2.12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th><strong>x→eng</strong></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean bitext</td>
<td>34.01</td>
<td>45.31</td>
<td>42.14</td>
<td>55.14</td>
<td>45.58</td>
<td>30.56</td>
<td>43.62</td>
<td>30.55</td>
</tr>
<tr>
<td>Noisy bitext</td>
<td>30.04</td>
<td>34.18</td>
<td>33.04</td>
<td>54.34</td>
<td>43.51</td>
<td>16.23</td>
<td>46.30</td>
<td>8.92</td>
</tr>
<tr>
<td>Clean + Noisy bitext</td>
<td>30.53</td>
<td>35.75</td>
<td>33.69</td>
<td>54.66</td>
<td>44.23</td>
<td>15.90</td>
<td>46.37</td>
<td>10.91</td>
</tr>
<tr>
<td>Filtered only</td>
<td>36.18</td>
<td>41.71</td>
<td>36.94</td>
<td>54.79</td>
<td>51.64</td>
<td>18.85</td>
<td>51.14</td>
<td>10.86</td>
</tr>
<tr>
<td>Filtered + Clean bitext</td>
<td>36.56</td>
<td>43.19</td>
<td>40.44</td>
<td>56.65</td>
<td>51.25</td>
<td>20.33</td>
<td>50.77</td>
<td>23.44</td>
</tr>
<tr>
<td>Filtered only</td>
<td>36.64</td>
<td>44.32</td>
<td>41.44</td>
<td>56.60</td>
<td>52.98</td>
<td>21.88</td>
<td>51.43</td>
<td>25.22</td>
</tr>
</tbody>
</table>

Table 6: Performance of the multilingual model on the FLORES-101 devset, with the maximum BLEU per column in **bold**. x represents African languages.
We find that on FLORES-101, adding in noisy, unfiltered data improves the results over just using the true parallel data. On MAFAND-MT, however, it generally reduces the BLEU score significantly. For both datasets, adding appropriately filtered data results in the highest performance averaged over all the languages, although for some specific languages, just using true parallel data resulted in the best performance.

Our performance on the test set provided by the organizers (Adelani et al., 2022b) is shown in Table 7. Here we can see that our primary model, which was trained on the clean bitext as well as the filtered data (filtered using ALBERT-xlarge, \( t = 0.7 \)), significantly outperforms the model trained only on the clean bitext. We also see that our approach seems to have a larger performance gain when translating from African languages compared to translating to them.

### 7 Conclusion and Future Work

In this work, we used a sentence-pair classifier to classify parallel data as being aligned, or not. Using this approach, we filtered out a large portion of the original, noisy, data and fine-tuned existing large language models on this new data. Our results show that training on the filtered data significantly increases the performance of the models, resulting in improved translations. In particular, our approach outperforms (i) training only on clean data, (ii) training only on filtered data, and (iii) training on the original dataset, consisting of clean and noisy data. This provides additional evidence in favor of prioritizing data quality over quantity, as well as the need for more advanced noise detection systems.

<table>
<thead>
<tr>
<th>Models</th>
<th>eng-xk</th>
<th>fra-xk</th>
<th>x-eng</th>
<th>x-fra</th>
<th>Avg.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean + Noisy bitext</td>
<td>8.91</td>
<td>12.49</td>
<td>19.24</td>
<td>20.00</td>
<td>7.77</td>
</tr>
<tr>
<td>Clean + Noisy bitext</td>
<td>8.52</td>
<td>12.83</td>
<td>14.35</td>
<td>28.37</td>
<td>5.17</td>
</tr>
<tr>
<td>Clean only</td>
<td>8.00</td>
<td>13.64</td>
<td>15.67</td>
<td>28.67</td>
<td>2.53</td>
</tr>
</tbody>
</table>

Table 7: Performance of the multilingual model on the MAFAND-MT test set, with the maximum BLEU per column in bold. x represents African languages.

other models, possibly suggesting that F1 performance does not fully indicate the expected downstream performance on the actual translation task.

### 6.2.1 The effect of filtering on translation models

We fine-tune M2M-100 for multilingual translation on the filtered data, and as expected, our results (on average) demonstrate a considerable improvement when the translation model is trained on the filtered data rather than the original noisy texts. In particular, for many languages, training on the filtered data from ALBERT-xlarge with a threshold of 0.7 outperformed the model trained on just the noisy bitext with at least a BLEU point.

Furthermore, we compared the performance of the model trained only on the clean data and on only the filtered data. Just as we saw with the baseline system, on MAFAND-MT, the model trained on the clean bitext performed better than the model trained on the filtered bitext, and on FLORES-101, the reverse was true. These results again confirm the importance of the filtering approach and further supports the observation that NMT engines are less robust to noise as found by Khayrallah and Koehn (2018), especially for low-resource settings.

### 6.3 Clean vs. filtered data

We find that on FLORES-101, adding in noisy, unfiltered data improves the results over just using the true parallel data. On MAFAND-MT, however, it generally reduces the BLEU score significantly.

### Table 7: Performance of the multilingual model on the MAFAND-MT test set, with the maximum BLEU per column in bold. x represents African languages.
We would also like to understand the reasons behind the performance by analyzing the filtered data more in depth. Finally, given more computational resources, we will (i) train the classifier for more epochs, using other language models and/or using different quality thresholds, (ii) use longer sentence length than the current 128, (iii) train the translation models on AfroXLMR and ALBERT-base filtered data, and (iv) use the filtering approach on more languages, to evaluate its generalizability. Ultimately, we hope that this filtering approach could lead to the use of cleaner data to train translation models, improving the overall translation quality for low-resources languages.

Acknowledgements

Idris Abdulmumin acknowledges funding from Google through the Africa Travel Grants. Computations were performed using High Performance Computing infrastructure provided by the Mathematical Sciences Support unit at the University of the Witwatersrand, LIAAD INESC TEC, Portugal, and DFKI GmbH. We acknowledge support from the Machine Learning Group, EISLAB, Luleå University of Science and Technology, Sweden. In addition, we thank the organizers for the compute grant, which allowed us to run some of our experiments. Finally, we thank Dr. Rachel Bawden, and the anonymous reviewers of the 2022 WMT shared task for their helpful feedback. We also acknowledge the useful inputs of Julia Kreutzer and Angela Fan.

References


Idris Abdulmumin, Bashir Shehu Galadanci, Shamsudeen Hassan Muhammad, and Garba Aliyu. 2022. Quantity vs. quality of monolingual source data in automatic text translation: Can it be too little if it is too good? In 2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON), pages 1–5. IEEE.


1010


A Appendix - Data Sources

Datasets used in this project and their sources, as listed in Table 9: MAFANDMT, wmt22_african, LAVA Corpus, XLEnt, Tanzil, WikiMatrix, CCAligned, CCMatrix, GlobalVoices, ParaCrawl, GNOME, tico-19, ELRC_2922, EUbookshop, KDE4, TED2020, Tatoeba, Ubuntu, bible-uedin, wikimedia, QED, MultiCCAligned and Mozilla-I10n.

---

9 https://drive.google.com/drive/folders/179AkJOP3fZMFS0rIyEBBDZ-WICs2wPWU
10 https://casmacat.eu/corpus/global-voices.html
11 https://globalvoices.org/
12 https://paracrawl.eu/
13 https://l10n.gnome.org/
15 https://elrc-share.eu/repository/browse/covid-19-health-wikipedia-dataset-multilingual-53-en-x-language-pairs/fe23e2c28c8311ea913100155d0267066f62c6b30ac0429f8d497df0abd2ef72/
16 http://bookshop.europa.eu
17 http://www.lt-innovate.org/lt-observe/resources/kde4-kde4-localization-files-v2
18 https://tatoeba.org/en/
19 https://translations.launchpad.net/
20 https://dumps.wikimedia.org/other/contesttranslation/
## Table 9: Training Data Used — TP=True Parallel; AA=Automatically Aligned

<table>
<thead>
<tr>
<th>Data</th>
<th>en</th>
<th>fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAFAND-MT</td>
<td>3,098</td>
<td>6,998</td>
</tr>
<tr>
<td>Tanzil</td>
<td>128,376</td>
<td>-</td>
</tr>
<tr>
<td>GlobalVoices</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>tico-19</td>
<td>3,071</td>
<td>-</td>
</tr>
<tr>
<td>ELRC_2922</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>EUbookshop</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Tatoeba</td>
<td>37</td>
<td>22</td>
</tr>
<tr>
<td>bible-uedin</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>QED</td>
<td>124</td>
<td>12</td>
</tr>
<tr>
<td>Mozilla-110n</td>
<td>4,952</td>
<td>4,172</td>
</tr>
<tr>
<td>Total (TP)</td>
<td>139,678</td>
<td>11,204</td>
</tr>
</tbody>
</table>

### Automatically Aligned

<table>
<thead>
<tr>
<th>Data</th>
<th>en</th>
<th>fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>WMT22 African</td>
<td>2,309,758</td>
<td>172,973</td>
</tr>
<tr>
<td>WebCrawl Afr.</td>
<td>16,950</td>
<td>3,372</td>
</tr>
<tr>
<td>LAVA Corpus</td>
<td>-</td>
<td>20,993</td>
</tr>
<tr>
<td>WikiMatrix</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CCAAligned</td>
<td>339,178</td>
<td>148,147</td>
</tr>
<tr>
<td>CCMatrix</td>
<td>5,861,080</td>
<td>80,385</td>
</tr>
<tr>
<td>ParaCrawl</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>GNOME</td>
<td>5,466</td>
<td>23,767</td>
</tr>
<tr>
<td>KDE4</td>
<td>1,493</td>
<td>-</td>
</tr>
<tr>
<td>TED2020</td>
<td>27</td>
<td>210</td>
</tr>
<tr>
<td>XLEnt</td>
<td>436,602</td>
<td>69,820</td>
</tr>
<tr>
<td>Ubuntu</td>
<td>242</td>
<td>635</td>
</tr>
<tr>
<td>wikimedia</td>
<td>23,385</td>
<td>12,279</td>
</tr>
<tr>
<td>MultiCCAligned</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Total (AA)</td>
<td>8,994,181</td>
<td>511,588</td>
</tr>
<tr>
<td>Total (ALL)</td>
<td>9,133,189</td>
<td>522,792</td>
</tr>
</tbody>
</table>

---

1014