Low-Resource Corpus Filtering using Multilingual Sentence Embeddings

Vishrav Chaudhary ♦ Yuqing Tang ♦ Francisco Guzmán ♦ Holger Schwenk ♦ Philipp Koehn

♦Facebook AI ♦ Johns Hopkins University
{vishrav, yuqtang, fguzman, schwenk}@fb.com phi@jhu.edu

Abstract

In this paper, we describe our submission to the WMT19 low-resource parallel corpus filtering shared task. Our main approach is based on the LASER toolkit (Language-Agnostic SEnnce Representations), which uses an encoder-decoder architecture trained on a parallel corpus to obtain multilingual sentence representations. We then use the representations directly to score and filter the noisy parallel sentences without additionally training a scoring function. We contrast our approach to other promising methods and show that LASER yields strong results. Finally, we produce an ensemble of different scoring methods and obtain additional gains. Our submission achieved the best overall performance for both the Nepali–English and Sinhala–English 1M tasks by a margin of 1.3 and 1.4 BLEU respectively, as compared to the second best systems. Moreover, our experiments show that this technique is promising for low and even no-resource scenarios.

1 Introduction

The availability of high-quality parallel training data is critical for obtaining good translation performance, as neural machine translation (NMT) systems are less robust against noisy parallel data than statistical machine translation (SMT) systems (Khayrallah and Koehn, 2018). Recently, there is an increased interest in the filtering of noisy parallel corpora (such as Paracrawl1) to increase the amount of data that can be used to train translation systems (Koehn et al., 2018).

While the state-of-the-art methods that use NMT models have proven effective in mining parallel sentences (Junczys-Dowmunt, 2018) for high-resource languages, their effectiveness has not been tested in low-resource languages. The implications of low availability of training data for parallel-scoring methods is not known yet.

For the task of low-resource filtering (Koehn et al., 2019), we are provided with a very noisy 40.6 million-word (English token count) Nepali–English corpus and a 59.6 million-word Sinhala–English corpus crawled from the web as part of the Paracrawl project. The challenge consists of providing scores for each sentence pair in both noisy parallel sets. The scores will be used to subsample sentence pairs that amount to 1 million and 5 million English words. The quality of the resulting subsets is determined by the quality of a statistical machine translation (Moses, phrase-based (Koehn et al., 2007)) and the neural machine translation system fairseq (Ott et al., 2019) trained on this data. The quality of the machine translation system will be measured by BLEU score using SacreBLEU (Post, 2018) on a held-out test set of Wikipedia translations for Sinhala–English and Nepali–English from the flores dataset (Guzmán et al., 2019).

In our submission for this shared task, we use of multilingual sentence embeddings obtained from LASER2 which uses an encoder-decoder architecture to train a multilingual sentence representation model using a relatively small parallel corpus. Our experiments demonstrate that the proposed approach outperforms other existing approaches. Moreover we make use of an ensemble of multiple scoring functions to further boost the filtering performance.

1http://www.paracrawl.eu/
2https://github.com/facebookresearch/LASER
2 Methodology

The WMT 2018 shared task for parallel corpus filtering (Koehn et al., 2018) introduced several methods to tackle a high-resource German-English data condition. While many of these methods were successful to filter out noisy translations, few have been tried under low-resource conditions. In this paper, we address the problem of low-resource sentence filtering using sentence-level representations and compare them to other popular methods used in high-resource conditions.

The LASER model (Artetxe and Schwenk, 2018a) makes use of multilingual sentence representations to gauge the similarity between the source and the target sentence. It has provided state-of-the-art performance on the BUCC corpus mining task and has also been effective in filtering WMT Paracrawl data (Artetxe and Schwenk, 2018a). However, these tasks only considered high-resource languages, namely French, German, Russian, and Chinese. Fortunately, this technique has also been effective on zero-shot cross-lingual natural language inference in the XNLI dataset (Artetxe and Schwenk, 2018b) which makes it promising for the low resource scenario focused in this shared task. In this paper, we propose to use an adaptation of LASER to low-resource conditions to compute the similarity scores to filter out noisy sentences.

For comparison to LASER, we also establish initial benchmarks using Bicleaner and Zipporah, two popular baselines which have been used in the Paracrawl project; and dual conditional cross-entropy, which has proven to be state-of-the-art for the high-resource corpus filtering task (Koehn et al., 2018). We explore the performance of the techniques under similar pre-processing conditions regarding language identification filtering and lexical overlap. We observe that LASER scores provide a clear advantage for this task. Finally, we perform ensembling of the scores coming from different methods. We observe that when LASER scores are included in the mix, the boost in performance is relatively minor. In the rest of this section we discuss the settings for each of the methods applied.

2.1 LASER Multilingual Representations

The underlying idea is to use the distances between two multilingual representations as a notion of parallelism between the two embedded sentences (Schwenk, 2018). To do this, we first train an encoder that learns to produce a multilingual, fixed-size sentence representation; and then compute a distance between two sentences in the learned embedding space. In addition, we use a margin criterion, which uses a $k$ nearest neighbors approach to normalize the similarity scores given that cosine similarity is not globally consistent (Artetxe and Schwenk, 2018a).

**Encoder** The multilingual encoder consists of a bidirectional LSTM, and our sentence embeddings are obtained by applying max-pooling over its output. We use a single encoder and decoder in our system, which are shared by all languages involved. For this purpose, we trained multilingual sentence embeddings on the provided parallel data only (see Section 3.2 for details).

**Margin** We follow the definition of ratio$^4$ from (Artetxe and Schwenk, 2018a). Using this, the similarity score between two sentences $(x, y)$ can be computed as

$$\frac{2k \cos(x, y)}{\sum_{y' \in \text{NN}_k(x)} \cos(x, y') + \sum_{x' \in \text{NN}_k(y)} \cos(x', y)}$$

where $\text{NN}_k(x)$ denotes the $k$ nearest neighbors of $x$ in the other language, and analogously for $\text{NN}_k(y)$. Note that this list of nearest neighbors does not include duplicates, so even if a given sentence has multiple occurrences in the corpus, it would have (at most) one entry in the list.

**Neighborhood** Additionally, we explored two ways of sampling $k$ nearest neighbors. First a global method, in which we used the neighborhood comprised of the noisy data along with the clean data. Second a local method, in which we only scored the noisy data using the noisy neighborhood, or the clean data using the clean neighborhood.$^5$

---

$^3$http://statmt.org/wmt18/parallel-corpus-filtering.html

$^4$We explored the absolute, distance and ratio margin criteria, but the latter worked best

$^5$This last part was only done for training an ensemble
2.2 Other Similarity Methods

**Zipporah** (Xu and Koehn, 2017; Khayrallah et al., 2018), which is often used as a baseline comparison, uses language model and word translation scores, with weights optimized to separate clean and synthetic noise data. In our setup, we trained Zipporah models for both language pairs Sinhala–English and Nepali–English. We used the open source release\(^6\) of the Zipporah tool without modifications. All components of the Zipporah model (probabilistic translation dictionaries and language models) were trained on the provided clean data (excluding the dictionaries). Language models were trained using KenLM (Heafield et al., 2013) over the clean parallel data. We are not using the provided monolingual data, as per default setting. We used the development set from the *flores* dataset for weight training.

**Bicleaner** (Sánchez-Cartagena et al., 2018) uses lexical translation and language model scores, and several shallow features such as: respective length, matching numbers and punctuation. As with Zipporah, we used the open source Bicleaner\(^7\) toolkit unmodified out-of-the-box. Only the provided clean parallel data was used to train this model. Bicleaner uses a rule-based component to identify noisier examples in the parallel data and trains a classifier to learn how to separate them from the rest of the training data. The use of language model features is optional. We only used models without a language model scoring component.\(^8\)

**Dual Conditional Cross-Entropy** One of the best performing methods on this task was dual conditional cross-entropy filtering (Junczys-Dowmunt, 2018), which uses a combination of forward and backward models to compute a cross-lingual similarity score. In our experiments, for each language pair, we used the provided clean training data to train neural machine translation models in both translation directions: source-to-target and target-to-source. Given such a translation model \(M\), we force-decode sentence pairs \((x, y)\) from the noisy parallel corpus and obtain the cross-entropy score

\[
H_M(y|x) = \frac{1}{|y|} \sum_{t=1}^{|y|} \log p_M(y_t|y_{[1,t-1]}, x) \tag{1}
\]

Forward and backward cross entropy scores, \(H_F(y|x)\) and \(H_B(x|y)\) respectively, are then averaged with an additional penalty on a large difference between the two scores \(|H_F(y|x) - H_B(x|y)|\).

\[
\text{score}(x, y) = \frac{H_F(y|x) + H_B(x|y)}{2} - |H_F(y|x) - H_B(x|y)| \tag{2}
\]

The forward and backward models are five-layer encoder/decoder transformers trained using *fairseq* with parameters identical to the ones used in the baseline *flores* model\(^9\). The models were trained on the clean parallel data for 100 epochs. For the Nepali-English task, we also explored using Hindi-English data without major differences in results. We used the *flores* development set to pick the model that maximizes BLEU scores.

2.3 Ensemble

To leverage over the strengths and weaknesses of different scoring systems, we explored the use of a binary classifier to build an ensemble. While it’s trivial to obtain positives (e.g. the clean training data), mining negatives can be a daunting task. Hence, we use positive-unlabeled (PU) learning (Mordelet and Vert, 2014), which allows us to obtain classifiers without having to curate a dataset of explicit positive and negatives. In this setting our positive labels come from the clean parallel data while the unlabeled data comes from the noisy set.

To achieve this, we apply bagging of 100 weak, biased classifiers (i.e. with a 2:1 bias for unlabeled data vs. positive label data). We use support vector machines (SVM) with a radial basis kernel, and we randomly sub-sample the set of features for training each base classifier, helping keep them diverse and low-capacity.

We ran two iterations of training of this ensemble. In the first iteration we used the original positive and unlabeled data described above. For the second iteration, we used the learned classifier to re-label the training data. We explored several re-labeling approaches (e.g. setting a threshold that maximizes \(F_1\) score). However, we found that setting a class boundary to preserve the original positives-to-unlabeled ratio worked best. We also observed that the performance deteriorated after two iterations.

\(^6\)https://github.com/hainan-xv/zipporah

\(^7\)https://github.com/bitextor/bicleaner

\(^8\)We found that including a LM as a feature resulted in almost all sentence pairs receiving a score of 0.

\(^9\)https://github.com/facebookresearch/flores#train-a-baseline-transformer-model
3 Experimental Setup

We experimented with various methods using a setup that closely mirrors the official scoring of the shared task. All methods are trained on the provided clean parallel data (see Table 1). We did not use the given monolingual data. For development purposes, we used the provided flores dev set. For evaluation, we trained machine translation systems on the selected subsets (1M, 5M) of the noisy parallel training data using fairseq with the default flores training parameter configuration. We report SacreBLEU scores on the flores devtest set. We selected our main system based on the best scores on the devtest set for the 1M condition.

<table>
<thead>
<tr>
<th>Sentences</th>
<th>en-ne</th>
<th>en-hi</th>
<th>hi-en</th>
</tr>
</thead>
<tbody>
<tr>
<td>English words</td>
<td>3.7M</td>
<td>3.7M</td>
<td>20.7M</td>
</tr>
</tbody>
</table>

Table 1: Available bitexts to train the filtering approaches.

3.1 Preprocessing

We applied a set of filtering techniques similar to the ones used in LASER (Artetxe and Schwenk, 2018a) and assigned a score of $-1$ to the noisy sentences based on incorrect language on either the source or the target side or having an overlap of at least 60% between the source and the target tokens. We used fastText\(^{10}\) for language id filtering. Since LASER computes similarity scores for a sentence pair using these filtering techniques, we experimented by adding these to the other models we used for this shared task.

3.2 LASER Encoder Training

For our experiments and the official submission, we trained a multilingual sentence encoder using the permitted resources in Table 1. We trained a single encoder using all the parallel data for Sinhala–English, Nepali–English and Hindi–English. Since Hindi and Nepali share the same script, we concatenated their corpora into a single parallel corpus. To account for the difference in size of the parallel training data, we over-sampled the Sinhala–English and Nepali/Hindi-English bitexts in a ratio of 5:3. This resulted in roughly 3.2M training sentences for each language direction, i.e. Sinhala and combined Nepali-Hindi.

The models were trained using the same setting as the public LASER encoder which involves normalizing texts and tokenization with Moses tools (falling back to the English mode). We first learn a joint 50k BPE vocabulary on the concatenated training data using fastBPE\(^{11}\). The encoder sees Sinhala, Nepali, Hindi and English sentences at the input, without having any information about the current language. This input is always translated into English.\(^{12}\) We experimented with various techniques to add noise to the English input sentences, similar to what is used in unsupervised neural machine translation, e.g. (Artetxe et al., 2018; Lample et al., 2018), but this did not improve the results.

The encoder is a five-layer BLSTM with 512 dimensional layers. The LSTM decoder has one hidden layer of size 2048, trained with the Adam optimizer. For development, we calculate similarity error on the concatenation of the flores dev sets for Sinhala–English and Nepali–English. Our models were trained for seven epochs for about 2.5 hours on 8 Nvidia GPUs.

4 Results

From the results in Table 2, we observe several trends: (i) the scores for the 5M condition are generally lower than for the 1M condition. This condition appears to be exacerbated by the application of language id and overlap filtering. (ii) LASER shows consistently good performance. The local neighborhood works better than the global one. In that setting, LASER is on average 0.71 BLEU above the best non-LASER system. These gaps are higher for the 1M condition (0.94 BLEU). (iii) The best ensemble configuration provides small improvements over the best LASER configuration. For Sinhala–English the best configuration includes every other scoring method (ALL). For Nepali–English the best configuration is an ensemble of LASER scores. (iv) Dual cross entropy shows mixed results. For Sinhala–English, it only works once the language id filtering is enabled which is consistent with previous observations (Junczys-Dowmunt, 2018). For Nepali–English, it provides scores well below the rest of the scoring methods. Note that we did not perform an architecture exploration.

\(^{10}\)https://fasttext.cc/docs/en/language-identification.html

\(^{11}\)https://github.com/glample/fastBPE

\(^{12}\)This means that we have to train an English auto-encoder. This didn’t seem to hurt, since the same encoder also handles the three other languages.
Table 2: SacreBLEU scores on the flores devtest set. In **bold**, we highlight the best scores for each condition. In *italics*+, we highlight the runner up. We also signal the best non-LASER method with +.

<table>
<thead>
<tr>
<th>Method</th>
<th>ne-en</th>
<th>si-en</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Zipporah</strong></td>
<td>1M</td>
<td>5M</td>
</tr>
<tr>
<td>base</td>
<td>5.03</td>
<td>2.09</td>
</tr>
<tr>
<td>+ LID</td>
<td>5.30</td>
<td>1.53</td>
</tr>
<tr>
<td>+ Overlap</td>
<td>5.35</td>
<td>1.34</td>
</tr>
<tr>
<td><strong>Dual X-Ent.</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>base</td>
<td>2.83</td>
<td>1.88</td>
</tr>
<tr>
<td>+ LID</td>
<td>2.19</td>
<td>0.82</td>
</tr>
<tr>
<td>+ Overlap</td>
<td>2.23</td>
<td>0.91</td>
</tr>
<tr>
<td><strong>Bicleaner</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>base</td>
<td>5.91</td>
<td></td>
</tr>
<tr>
<td>+ LID</td>
<td>5.88</td>
<td>2.09</td>
</tr>
<tr>
<td>+ Overlap</td>
<td>6.12+</td>
<td>2.14</td>
</tr>
<tr>
<td><strong>LASER</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td><em>local</em></td>
<td>7.37*</td>
<td>3.15</td>
</tr>
<tr>
<td><em>global</em></td>
<td>6.98</td>
<td>2.98*</td>
</tr>
<tr>
<td><strong>Ensemble</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ALL</td>
<td>6.17</td>
<td>2.53</td>
</tr>
<tr>
<td>LASER glob. + loc.</td>
<td>7.49</td>
<td>2.76</td>
</tr>
</tbody>
</table>

Table 3: Official results of the main and secondary submissions on the flores test set evaluated with the NMT configuration. For comparison, we include the best scores by another system.

<table>
<thead>
<tr>
<th>Method</th>
<th>ne-en</th>
<th>si-en</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Main - Ensemble</strong></td>
<td>1M</td>
<td>5M</td>
</tr>
<tr>
<td>Constr. - LASER <em>local</em></td>
<td>6.8</td>
<td>2.8</td>
</tr>
<tr>
<td>Best (other)</td>
<td>5.5</td>
<td>3.4</td>
</tr>
</tbody>
</table>

Table 4: Comparison of results on the flores devtest set using the constrained and the pre-trained versions of LASER.
References


