Score Combination for Improved Parallel Corpus Filtering for Low Resource Conditions

Muhammad ElNokrashy [1], **Amr Hendy**[1], **Mohamed Abdelghaffar**[1], **Mohamed Afify**[1], **Ahmed Tawfik**[1] **and Hany Hassan Awadalla**[2]
[1] Microsoft Egypt Development Center, Cairo, Egypt
[2] Microsoft Corporation, Redmond, WA, USA
{muhammad.elnokrashy,a-amhend,mohamed.abdelghaar,mafify,atawfik,hanyh}@microsoft.com

Abstract

This paper presents the description of our submission to WMT20 sentence filtering task. We combine scores from custom LASER built for each source language, a classifier built to distinguish positive and negative pairs and the original scores provided with the task. For the mBART setup, provided by the organizers, our method shows 7% and 5% relative improvement, over the baseline, in sacreBLEU score on the test set for Pashto and Khmer respectively.

1 Introduction

Neural machine translation (NMT) brings significant gains to the field of machine translation. However, it is known to be very sensitive to the quality of parallel data (Khayrallah and Koehn, 2018). This becomes a serious problem when using large but very noisy corpora for training. There is a lot of work on filtering noisy parallel data. For example, the work in (Junczys-Dowmunt, 2018) provides excellent results for large corpora. Low resource languages are even more challenging and the results in (Chaudhary et al., 2019) using multilingual embeddings are very encouraging.

This paper describes the system submitted to the WMT20 Shared Task on Parallel Corpus Filtering for Low Resource Conditions. Due to time limitation our submission covers only sentence pair filtering. However, some of the proposed techniques could be used for sentence alignment and filtering. The task focuses on the Pashto-English and Khmer-English language pairs. It is required that the participants calculate scores to sort very noisy parallel sentence pairs provided for each language. The top scoring pairs leading to 5M tokens on the English side are then used to train machine translation for each language pair. The organizers also provide LASER-based scores as a baseline according to the method in (Chaudhary et al., 2019), with possibly some modifications. We describe our general system architecture followed by development experiments to evaluate the merit of different methods and finally report the performance of our best setup per language, using the fairseq recipe provided for the task for both full training (from scratch), and finetuning (mBART) settings. mBART (Liu et al., 2020) is a recently proposed pretraining method. Models initialized using mBART for both language pairs are provided with the task.

Our proposed approach combine the scores provided by the organizers with the following two scores:

- Margin distance calculated based on custom LASER built for each language using parallel data and a large amount of forward translated mono data provided by the organizers.
- Cosine distance between the embeddings generated by a classifier taking the pretrained LASER as input and trained to distinguish parallel and non-parallel sentence pairs using parallel data provided in the task.

More details on the approach are provided in Section 2. For the mBART setup our method shows 7% and 5% relative improvement in sacreBLEU score on the test set for Pashto and Khmer respectively.

2 System Architecture

In this section we describe the overall system architecture. We start by presenting language-based preprocessing in Section 2.1.The scores provided by the organizers use pretrained LASER (Artetxe and Schwenk, 2019b) and margin distance (Artetxe and Schwenk, 2019a; Johnson et al., 2017). Actually their method obtained state-of-the-art results in WMT19 low resource filtering task for Sinhala,

Proceedings of the 5th Conference on Machine Translation (WMT), pages 947–951
Online, November 19–20, 2020.

Nepali and Hindi and hence is a very strong baseline. Therefore, we opt to construct sentence embeddings using two different methods, namely custom LASER and classifier-based embeddings, to complement the baseline pretrained LASER. Each embedding method is then used to generate a score for each sentence pair and all the scores are combined to form the final score. In the rest of this section, we will describe custom LASER in Section 2.2 followed by classifier-based embeddings in Section 2.3 and finally we outline score combination in Section 2.4.

2.1 Preprocessing

We first preprocess the data using language identification on the source side. The results reported in this paper use the BLING (BLING, 2020) tool from Microsoft but preliminary experiments indicate similar performance using python langid or fasttext (Bojanowski et al., 2017). One interesting feature of BLING is that it returns the percentages of different language constituents of a sentence. In the current implementation we keep sentences that have a language identification score of the source language of 80% or higher. All sentences that do not satisfy the language identification threshold are assigned a very low score and hence are not selected in the final candidate pairs. We will report results without using language identification in the experiments. We generally found it helps a lot for Khmer and actually hurts a bit for Pashto. We expect this is due to the larger Khmer size. Table 1 shows the original number of sentences and those filtered at 80% threshold for Pashto and Khmer.

Language	Original	Filtered (kept)
Ps	1,022,883	615,451 (60.17%)
Km	4,169,574	2,714,664 (65.11%)

Table 1: Number of sentences before and after language filtering

2.2 Custom LASER

Pretrained LASER (Artetxe and Schwenk, 2019b) has 93 languages. Some of these languages are under-represented and others, like Pashto, are completely missing. While similar languages tend to help each other it is clearly beneficial to have a custom LASER trained for the languages of interest. In WMT19 results (Chaudhary et al., 2019), custom LASER trained on a combination of Hindi, Sinhala and Nepali outperformed the pretrained LASER for the filtering task. Here, we build two separate models for Pashto and Khmer. Since both languages have very different origins we thought it is not beneficial to build a combined model but we haven't verified this experimentally. We use the LASERtrain package (Esplà and Sánchez-Martínez, 2019) to train the custom LASER. This package follows the LASER training as given in (Artetxe and Schwenk, 2019b) and provides experiments on BUCC'18 with good results. It is also possible to fine-tune the pretrained LASER using the languages of interest. We will explore this in future work.

Participants in WMT20 are limited to data provided by the organizers. The supplied parallel data for both languages is of rather limited size and is dominated by domain specific data as software localization and religious text and hence we use the provided monolingual text to augment the training data. For each language this is done as follows. We start with the provided sentence scores and filter 5M English tokens as suggested in the task. We use the resulting parallel data to train an mBART initialized MT system. The sacreBLEU scores on the development test set from the organizers and our internal run are shown in Table 2.

Language-pair	Organizers	Internal
Ps-En	12.2	11.6
Km-En	10.6	10.4

Table 2: SacreBLEU for mBART on development test set as provided by the organizers and for internal run.

In addition to the noisy parallel sentences, the organizers provide additional parallel data and monolingual data for both languages. The parallel data comes mainly from OPUS and consists of 290K and 123K pairs for Khmer and Pashto respectively. The monolingual data for Khmer has around 13M sentences while that of Pashto has around 6M sentences. For more information about the sources of these data we refer the reader to (EMNLP, 2020). The resulting internal system is used to forward translate various amounts of monolingual data from the source language into English. We found in preliminary experiments that using around 3M monolingual sentences gives good performance (more

on the evaluation below). These sentences are randomly selected from the monoingual data. The synthetic pairs for Pashto-English and Khmer-English are used to augment the provided clean parallel data to train the custom LASER for each language. The reason we build unidirectional custom LASER is that most of our data is synthetic and the performance of the translation in the opposite direction English-Pashto(Khmer) is expected to be quite poor. In addition, English is very well represented in the pretrained LASER.

It is good to have a way to evaluate the quality of the built custom LASER. Building machine translation (MT) every time is very costly. To this end, we use a BUCC-like setup to evaluate the quality of the custom embeddings. We use the development set for each language pair. For each sentence on the source side we use the corresponding embedding to find the nearest neighbor, based on cosine distance, on the target side and calculate the top-1 accuracy. We do the same in the other direction (target to source) and average both numbers.

$$\text{BUCC}(S) = \frac{1}{2|S|}\left(\sum_i \mathbb{I}\left(\arg\max_j S_{i,j} = i\right) + \sum_j \mathbb{I}\left(\arg\max_i S_{i,j} = j\right)\right) \quad (1)$$

where S is the matrix of pairwise similarity scores for pairs in the source-target development set. Accuracies using pretrained and custom LASER for Pashto and Khmer are shown in Table 3.

Language-pair	Pretrained	Custom
Ps-En	9.56%	31.97%
Km-En	1.04%	39.50%

Table 3: BUCC-like accuracy scores on devtest set of the filtration task

Once the custom LASER of a language is trained it is used to calculate the score of a sentence pair using the margin distance as shown in Equation 2. The margin is implemented efficiently using (Johnson et al., 2017).

$$\begin{aligned}\text{score}(x, y) = &\text{margin}(\cos(x, y), \\ &0.5\,(\text{mean}\,\{\cos(z, x) \mid z \in \text{NN}_k(x)\} \\ &+ \text{mean}\,\{\cos(z, y) \mid z \in \text{NN}_k(y)\}))\end{aligned} \quad (2)$$

2.3 Classifier-Based Scores

In addition to custom LASER presented in the previous section we use scores provided from a classifier trained to distinguish parallel and non-parallel sentence pairs. It takes pretrained LASER embeddings of a sentence pair u and v and transforms them using a fully-connected layer with ReLU nonlinearity. Similar to (Reimers and Gurevych, 2019) it inputs the concatenation $[u_{tr}; v_{tr}; |u_{tr} - v_{tr}|]$, where u_{tr} and v_{tr} are the outputs of the fully connected layer, to a softmax classifier with two outputs representing the positive and negative pairs. The network is trained using the cross-entropy criterion. During testing, LASER embeddings of a sentence pair are passed through the fully connected layer and their cosine distance is calculated as the required score. The rationale is that the transformed embeddings provide better representation to separate positive and negative pairs compared to pretrained LASER.

For each language, the classifier is trained on the positive pairs provided by the organizers. Following (Zhang et al., 2020) for each sentence the negative pair is selected at random from the following:

- Select a sentence from its adjacent sentences within a window size of k (where k = 2 in our experiments).
- Truncate 30-70% words of the sentence.
- Swap the order of 30-70% of the words of the sentence.

After forming the positive and negative data around 500 example pairs, per language, are kept as validation set. The classification accuracy, on the validation set, for Pashto is 97% while that of Khmer is 98.5%.

2.4 Score Combination

Based on the previous sections each input sentence pair x, y has three scores. Assume the pretrained LASER embeddings are x_p and y_p and the custom

LASER embeddings are x_c and y_c. We can write the combined score $S(x, y)$ as follows:

$$S(x, y) = S_{mg}(x_p, y_p) + S_{mg}(x_c, y_c) + S_{cl}(x_p, y_p) \quad (3)$$

where $S_{mg}()$ indicates margin distance and $S_{cl}()$ indicates classifier distance. We choose to use a simple sum instead of using trainable weights because the provided parallel data that could be used to train the weights is very specific and could result in biased estimates of the weights.

We also experiment with minimum-maximum normalization that we found very useful in the case of Khmer. For this normalization each component score in Equation 3 is modified as follows:

$$S_{norm} = \frac{S - S_{min}}{S_{max} - S_{min}} \quad (4)$$

where S_{min} and S_{max} are the minimum and maximum scores over all the pairs.

3 Experimental Results

In this section we first present the results of various experiments to arrive at the final system architecture for both languages in Section 3.1. An internal system with a small architecture is used for fast turn-around. This is followed by running experiments with the final architecture using the official scripts provided by the organizers for both the from scratch and mBART settings.

3.1 Development Experiments

This section outlines various development experiments for Pashto and Khmer. As mentioned above an internal system with a small configuration is used to compare different configurations. The sacreBLEU scores for Pashto are shown in Table 4 while those for Khmer are in Table 5.

	Dev. Set	Test Set
B	7.4	8.4
C	9.3	9.3
B + C	9.2	10.4
B + C + Cl	9.5	10.5
B + C + Cl (BL)	8.3	9.6

Table 4: Pashto Development results (in SacreBLEU) for different configurations on development and test sets. B stands for baseline, C for custom, Cl for classifier and BL for BLING

	Dev. Set	Test Set
B	8.8	7.0
C	4.3	3.6
C (BL)	6.5	5.2
B + C (BL)	9.5	7.6
B + C + Cl (BL)	9.9	8

Table 5: Khmer Development results (in SacreBLEU) for different configurations on development and test sets.B stands for baseline, C for custom, Cl for classifier and BL for BLING.B+C+Cl result for Khmer uses minimum-maximum normalization.

From the two tables we can see that there is some significant difference in behavior between Pashto and Khmer. This can be summarized as follows:

- While custom LASER is better than pretrained LASER for Pashto it is worse for Khmer. We attribute this to the existence of Khmer and absence of Pashto in pretrained LASER. For Pashto, even with some small parallel data and synthetic data we can see some nice gains.
- BLING language filtering is crucial for Khmer while it hurts a bit for Pashto. We attribute this to the larger size and the noisier nature (from the view point of having more English words in the source side) of the Khmer data.
- Even if the custom LASER for Khmer is significantly worse than the pretrained one it helps when combined with the baseline.

3.2 Final Experiments

Based on the above observations, we decided to have our configurations for the final experiments of the two languages as follows:

- Use BLING filtering for Khmer but not for Pashto.
- Use the combined scores of pretrained LASER, custom LASER and classifier for both Pashto and Khmer.
- Experiment with and without min-max normalization.

The results using both from scratch (Full) and mBART (FT) settings as supplied by the organizers are shown in Table 6.

	Mode	Base	Ensemble	
			min-max	No norm
Ps-En	Full	10.04	10.12	10.05
	FT	11.61	11.99	12.38
Km-En	Full	7.16	7.88	6.36
	FT	10.56	11.12	9.02

Table 6: Final results in SacreBLEU on devtest set. Full stands for from Scratch and FT for mBART.

Based on the results in the table our submission used minimum-maximum normalization for Khmer but not for Pashto. By looking into the unnormalized scores we found that for Khmer they tend to be dominated by the classifier score, undermining both the baseline and custom LASER, and hence the normalization helps to bring all scores to the same dynamic range.

4 Summary

This paper present the description of our submission to WMT20 sentence filtering task. By building custom LASER and a classifier to distinguish positive and negative pairs. For the mBART setup our method shows 7% and 5% relative improvement in sacreBLEU score on the test set for Pashto and Khmer respectively.There are a lot of extensions along the proposed directions to improve sentence filtering for the low resource setting.

References

Mikel Artetxe and Holger Schwenk. 2019a. Margin-based Parallel Corpus Mining with Multilingual Sentence Embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3197–3203, Florence, Italy. Association for Computational Linguistics.

Mikel Artetxe and Holger Schwenk. 2019b. Massively multilingual sentence embeddings for zero-shot cross-lingual transfer and beyond. *Transactions of the Association for Computational Linguistics*, 7:597–610.

BLING. 2020. https://docs.microsoft.com/en-us/azure/cognitive-services/translator/.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Vishrav Chaudhary, Yuqing Tang, Francisco Guzmán, Holger Schwenk, and Philipp Koehn. 2019. Low-resource corpus filtering using multilingual sentence embeddings. *Proceedings of the Fourth Conference on Machine Translation (Volume 3: Shared Task Papers, Day 2)*.

EMNLP. 2020. Shared Task: Parallel Corpus Filtering and Alignment for Low-Resource Conditions.

Miquel Esplà and Felipe Sánchez-Martínez. 2019. LASERtrain. `https://github.com/transducens/LASERtrain/`.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2017. Billion-scale similarity search with gpus. *arXiv preprint arXiv:1702.08734*.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation, Volume 2: Shared Task Papers*, page 901–908, Belgium, Brussels.

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*, page 74–83, Melbourne, Australia.

Y. Liu, J. Gu, N. Goyal, X. Li, S. Edunov, M. Ghazvininejad, M. Lewis, and L. Zettlemeyer. 2020. Multilingual denoising pretraining for neural machine translation. *arXiv preprint arXiv:2001.08210v2*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3980–3990, Hong Kong, China. Association for Computational Linguistics.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel Corpus Filtering via Pre-trained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8545–8554, Online. Association for Computational Linguistics.