

Improving Machine Translation with Phrase Pair Injection and Corpus Filtering

Akshay Batheja, Pushpak Bhattacharyya

CFILT, Indian Institute of Technology Bombay
{akshaybatheja, pb}@cse.iitb.ac.in

Abstract

In this paper, we show that the combination of Phrase Pair Injection and Corpus Filtering boosts the performance of Neural Machine Translation (NMT) systems. We extract parallel phrases and sentences from the pseudo-parallel corpus and augment it with the parallel corpus to train the NMT models. With the proposed approach, we observe an improvement in the Machine Translation (MT) system for 3 low-resource language pairs, Hindi-Marathi, English-Marathi, and English-Pashto, and 6 translation directions by up to 2.7 BLEU points, on the FLORES test data. These BLEU score improvements are over the models trained using the whole pseudo-parallel corpus augmented with the parallel corpus.

1 Introduction

Deep Neural Architectures are *data hungry*. It is believed that the robustness of an NMT model depends on the size of the training corpus. However, not all language pairs have a substantial amount of parallel data. The primary data resource for low-resource languages is the web. However, the web-crawled data contains a lot of noise that degrades the performance of NMT systems. Hence, the quality of the training data is as important as its quantity. We aim to improve the quality of Machine Translation for Hindi-Marathi, English-Marathi, and English-Pashto language pairs by using the LaBSE-based (Feng et al., 2020) corpus filtering along with the Phrase Pair Injection. We use Phrase Pair Injection to increase the size and LaBSE-based corpus filtering to improve the quality of the parallel data extracted from the pseudo-parallel corpus. We observe that using these two techniques together makes the optimum use of the pseudo-parallel corpus.

The contributions of this work are as follows:

- We extract good quality parallel sentences and phrases from the pseudo-parallel corpus of

huge size. To the best of our knowledge, this is the first method that combines Phrase Pair Injection with neural-based Corpus Filtering and extracts both good quality parallel sentences and phrases from the pseudo-parallel corpus.

- We show that the extracted parallel sentences and phrases significantly improve the performance of the MT systems.

2 Related Work

Neural Networks have become very popular with increased computational power in recent times. The Transformer model introduced by Vaswani et al. (2017) gave significant improvements in the quality of translation as compared to the previous approaches (Bahdanau et al., 2015; Sutskever et al., 2014). Transformers allow parallelization, which enables the model to train faster and get better performances.

There are several prior studies to improve the quality and size of the parallel corpus. A set of heuristic rules were proposed by Lu et al. (2020) to remove low-quality sentence pairs from the noisy parallel corpus. Another popular approach is to compute cosine similarities between the source and target sentence embeddings. Feng et al. (2020) proposed the LaBSE model, which is a multilingual sentence embedding model trained on 109 languages, including some Indic languages. Pourmostafa Roshan Sharami et al. (2021) proposed a data selection pipeline that selects the In-Domain data from the Generic Domain based on its similarity with the other In-Domain data. This helps the MT model to perform well in domain-specific cases.

Sen et al. (2021) augmented the raw phrase table with the parallel corpus. The raw phrase table contains very noisy and repetitive phrase pairs. We observe that augmenting the whole phrase table

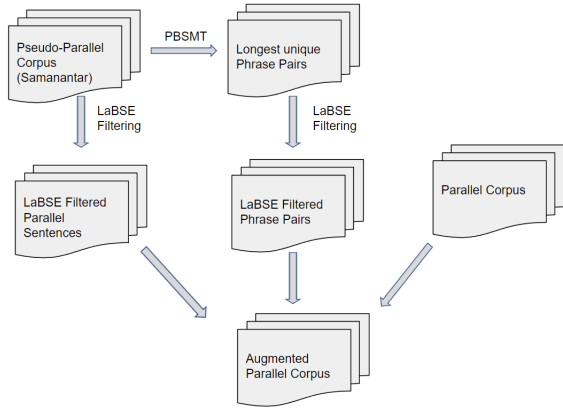


Figure 1: Phrase Pair Injection with LaBSE Filtering Pipeline

with parallel corpus does not show much improvement in the performance of the MT system. In contrast, we first extract the longest unique phrase pairs from the phrase table and then further filter them using LaBSE filtering to extract good quality phrase pairs. Then we augment these good quality phrase pairs with the LaBSE filtered parallel sentences. This helps improve the performance of MT systems significantly.

3 Approaches

We first discuss a method to extract good-quality parallel sentences from the pseudo-parallel corpus. Then we discuss a method to extract good quality phrase pairs from the pseudo-parallel corpus.

3.1 LaBSE based Filtering

In this approach, we aim to extract good-quality parallel sentences from the pseudo-parallel corpus. Language Agnostic BERT Sentence Embedding model (Feng et al., 2020) is a multilingual embedding model that supports 109 languages, including some Indic languages. We generate the sentence embeddings for the source and target sides of the pseudo-parallel corpora using the LaBSE¹ model. Then, we compute the cosine similarity between the source and target sentence embeddings. After that, we extract good quality parallel sentences based on a threshold value of the similarity scores. We calculate the average similarity score on a small dataset from the PM-India corpus (PMI) (Haddow and Kirefu, 2020). The PMI corpus consists of high-quality sentence pairs, so it helps us decide upon the threshold value.

¹<https://huggingface.co/sentence-transformers/LaBSE>

3.2 Phrase Pair Injection (PPI) with LaBSE Filtering

In this approach, we aim to extract good-quality parallel phrases from a noisy pseudo-parallel corpus. We first train a PBSMT model on the noisy pseudo-parallel corpus using the *Moses*² decoder, followed by tuning using MERT. Then, we extract phrase pairs from the generated Phrase Table based on their weighted average translational and lexical probabilities mentioned in the phrase table. As each sentence pair leads to multiple and repetitive phrase pairs in the Phrase Table, we only keep the longest unique phrase pairs from the extracted phrase pairs. Then, we perform the LaBSE-based filtering on these longest unique phrase pairs to remove the poor quality phrase pairs.

We augment these good-quality phrase pairs to the LaBSE filtered parallel sentences. This approach allows us to use the pseudo-parallel corpora at its full potential because we are extracting good quality parallel sentences as well as phrase pairs.

Corpus Name	Language Pairs	Sentence Pairs
Parallel Corpus	Hindi-Marathi	604K
	English-Marathi	248K
	English-Pashto	123K
Pseudo-Parallel Corpus	Hindi-Marathi	1.98M
	English-Marathi	3.28M
	English-Pashto	1.02M

Table 1: Dataset Statistics of Parallel and Pseudo-Parallel Corpus

4 Experimental Setup

In this section, we discuss the setup of various experiments that we performed. We use Byte Pair Encoding (Sennrich et al., 2016) as a segmentation technique to split words into subwords. We use 16000 merge operations for all experiments. We use the *OpenNMT-py*³ (Klein et al., 2017) library to train the Transformer based NMT models. We also train the PBSMT systems using Moses (Koehn et al., 2007). Then, we perform tuning using Minimum Error Rate Training (MERT) to find the optimal weights which maximize the translation performance. We use a tune set of the first 2000 parallel sentences from the pseudo-parallel corpus for the MERT tuning. Then, we extract the

²<http://www2.statmt.org/ Moses/?n=Development.GetStarted>

³<https://github.com/OpenNMT/OpenNMT-py>

Technique	# Sentence Pairs	Hindi→Marathi				Marathi→Hindi			
		WAT 2021		FLORES 101		WAT 2021		FLORES 101	
		BLEU	chrF	BLEU	chrF	BLEU	chrF	BLEU	chrF
Baseline+LaBSE +PPI- LaBSE	1.75M	17.6	51.0	10.0	42.5	25.4	51.5	16.2	42.5
Baseline+LaBSE +PPI	2.55M	17.4	51.0	10.2	43.0	25.3	51.6	16.5	42.5
Baseline+PPI-LaBSE	1.4M	15.5	49.8	9.5	42.0	23.4	50.2	15.4	41.6
Baseline+LaBSE	960K	17.5	50.3	9.4	41.9	25.1	51.2	15.9	41.6
Baseline+PPI	2.2M	16.1	49.9	9.2	41.4	23.1	49.9	15.1	40.8
No Filtering	2.56M	16.8	49.9	9.1	39.5	24.0	49.9	14.8	39.8
Baseline	604K	15.2	48.0	7.6	38.8	22.1	48.1	14.4	39.7

Table 2: BLEU and chrF scores of Hindi-Marathi NMT models.

Technique	# Sentence Pairs	English→Marathi		Marathi→English	
		FLORES 101		FLORES 101	
		BLEU	chrF	BLEU	chrF
Baseline + LaBSE + PPI-LaBSE	3.24M	9.8	40.3	17.0	44.6
Baseline + LaBSE + PPI	4.09M	9.9	40.5	16.2	43.4
Baseline + PPI-LaBSE	640K	6.2	33.8	12.7	40.3
Baseline + LaBSE	2.85M	8.8	39.8	16.7	44.0
Baseline + PPI	1.49M	6.6	35.0	12.7	40.3
No Filtering	3.53M	8.8	37.8	15.9	43.3
Baseline	248K	5.1	32.4	10.2	38.0

Table 3: BLEU and chrF scores of English-Marathi NMT models

phrase pairs from the generated Phrase Table, based on their weighted average translational and lexical probabilities mentioned in the phrase table. Further training details can be found in the appendix A.

4.1 Dataset Preparation

We train the Hindi-Marathi, English-Marathi, and English-Pashto NMT models using their respective Parallel and Pseudo-Parallel Corpus. The Parallel corpus for English-Marathi and Hindi-Marathi language pairs consists of the Indian Languages Corpora Initiative (ILCI) phase 1 corpus (Jha, 2010), BIBLE corpus (Christos Christodouloupoulos, 2015), Press Information Bureau corpus (Jerin Philip and Jawahar, 2021) (PIB), and PM-India corpus (PMI) (Haddow and Kirefu, 2020). The Hindi-Marathi Parallel corpus also consists of Tatoeba challenge dataset (Tiedemann, 2020). In the Hindi-Marathi Parallel corpus, except for ILCI, the parallel data was synthetically generated by translating the English sentences of the English-Marathi corpus to Hindi using Google Translation API. The Pseudo-Parallel corpus consists of Samanantar Corpus (Ramesh et al., 2021). For English-Pashto language pair, we use the parallel and pseudo-parallel corpus provided by the WMT20 shared task on Parallel Corpus Filtering and Alignment (Koehn et al., 2020). The detailed corpus statistics are mentioned in Table 1.

For evaluation, we use the test set introduced in WAT 2021 MultiIndicMT: An Indic Language

Multilingual Task and FLORES 101. The test set from WAT 2021 contains 2,390 sentences and is a part of the PMI corpus. The FLORES 101 test set contains 1012 sentences.

4.2 Baseline

We train the baseline models (Hindi-Marathi, English-Marathi and English-Pashto) directly on their respective parallel corpus.

4.3 No Filtering

In No Filtering model, we train the NMT models on the whole pseudo-parallel corpus augmented with parallel corpus.

4.4 Baseline + PPI

In this model, we first train a PBSMT model on the pseudo-parallel Corpus, followed by tuning using MERT. Then, we extract longest unique phrase pairs from the Phrase Table using a threshold Probability value of 0.95 for Hindi-Marathi and 0.8 for English-Marathi. Finally, we augment the parallel corpus with the extracted phrases pairs to train the NMT models.

4.5 Baseline + LaBSE

In this model, we use the LaBSE filtering technique with a threshold of **0.9**, **0.8** and **0.8** to extract 350K, 2.6M, and 166K good-quality parallel sentences from the Hindi-Marathi, English-Marathi

Technique	# Sentence Pairs	English→Pashto FLORES 101		Pashto→English FLORES 101	
		BLEU	chrF	BLEU	chrF
Baseline + LaBSE + PPI-LaBSE	342K	8.6	31	10.0	36.2
Baseline + LaBSE + PPI	790K	8.7	31	10.4	37.1
Baseline + PPI-LaBSE	176K	0.9	13.1	1.1	17.3
Baseline + LaBSE	290K	8.0	30.5	9.8	36.4
Baseline + PPI	624K	0.7	12.3	0.8	13.9
No Filtering	1.14M	6.0	23.2	9.4	34.4
Baseline	124K	0.2	8.7	0.4	15.9

Table 4: BLEU and chrF scores of English-Pashto NMT models

and English-Pashto pseudo-parallel corpus, respectively. Then, we augment the parallel corpus with the LaBSE filtered parallel sentences and train the respective NMT models.

4.6 Baseline + LaBSE + PPI

In this model, we combine the two techniques mentioned in the section 4.4 and 4.5. We first extract longest unique phrase pairs from pseudo-parallel corpus. Then, we extract good quality parallel sentences from pseudo-parallel corpus using LaBSE filtering. Finally, we augment the parallel corpus with the extracted longest unique phrase pairs and LaBSE filtered parallel sentences to train the NMT models.

4.7 Baseline + PPI-LaBSE

In this model, we first extract the longest unique phrase pairs using the technique mentioned in section 4.4. Then we apply LaBSE filtering on these extracted phrase pairs using a threshold value of 0.9. Finally, we augment the parallel corpus with the LaBSE filtered phrase pairs to train the NMT models.

4.8 Baseline + LaBSE + PPI-LaBSE

In this model, We combine the techniques mentioned in the section 4.5 and 4.7. We first extract LaBSE filtered parallel sentences from pseudo-parallel corpus. Then, we extract LaBSE filtered longest unique phrase pairs from the pseudo-parallel corpus. Finally, we augment the parallel corpus with the LaBSE filtered parallel sentences and LaBSE filtered phrase pairs.

5 Results and Analysis

We evaluate our NMT models using BLEU (Papineni et al., 2002) and chrF score metric. We use sacrebleu (Post, 2018) python library to calculate the BLEU and chrF scores. The results of all the experiments are summarized in Table 2, 4.

We can observe from the results that the models trained using the extracted phrase pairs and parallel sentences outperform the models trained using the whole pseudo-parallel corpus when augmented with the parallel corpus. The results of Hindi-Marathi, English-Marathi and English-Pashto MT models show that **Baseline + LaBSE + PPI-LaBSE** and **Baseline + LaBSE + PPI**, perform best amongst others on **WAT 2021** and **FLORES 101** test data, respectively.

In Hindi-Marathi MT, the best models improve the performance by **0.8** and **1.1** BLEU score points in *Hindi→Marathi* and **1.4** and **1.7** BLEU score points in *Marathi→Hindi* over the **No Filtering** model on FLORES 101 and WAT2021 test data.

In English-Marathi MT, the best models improve the performance by **1.0** and **1.1** BLEU score points in *English→Marathi* and *Marathi→English* over the **No Filtering** model, on the FLORES 101 test data.

In English-Pashto MT, the best models improve the performance by **2.7** and **1.0** BLEU score points over the **No Filtering** model, on the FLORES 101 test data.

We observe that even though the corpus size of the best models is smaller than the pseudo-parallel corpus, it still performs better as it has a higher proportion of good quality parallel sentences and phrase pairs.

6 Conclusion and Future Work

In this work, we show that extracting parallel sentences and phrase pairs from a pseudo-parallel corpus helps the NMT models improve its performance significantly when augmented with a parallel corpus. We show that LaBSE filtering assists the Phrase Pair Injection to extract the parallel data, which has good quality and quantity.

In the future, we plan to use the proposed corpus filtering techniques for other language pairs. This will provide us with a general overview of

how these filtering techniques perform on multiple languages. We also anticipate improvements in the result by trying different threshold values in the LaBSE filtering and Phrase Pair Injection.

7 Acknowledgements

We would like to thank the anonymous reviewers for their insightful feedback. We also express our gratitude towards Shivam Mhaskar, Sourabh Deoghare, Jyotsana Khatri, and other members of the Machine Translation group at CFILT, IIT Bombay, for their interesting and insightful comments.

8 Limitations

The size of extracted phrase pairs from the pseudo-parallel corpus is huge, and computing sentence embeddings using LaBSE for such a huge corpus is computationally expensive. We use various threshold values in LaBSE Filtering and Phrase Pair Injection. We need to try multiple combinations of these threshold values to obtain good results.

References

- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.
- Mark Steedman. Christos Christodouloupoulos. 2015. A massively parallel corpus: the bible in 100 languages. *Language resources and evaluation*.
- Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Ariavazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding.
- Barry Haddow and Faheem Kirefu. 2020. Pmindia – a collection of parallel corpora of languages of india.
- Vinay P Namboodiri Jerin Philip, Shashank Siripragada and CV Jawahar. 2021. Revisiting low resource status of indian languages in machine translation. In *8th ACM IKDD CODS and 26th COMAD*.
- Girish Nath Jha. 2010. The TDIL program and the Indian language corpora initiative (ILCI). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. European Language Resources Association (ELRA).
- Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*. Association for Computational Linguistics.
- Philipp Koehn, Vishrav Chaudhary, Ahmed El-Kishky, Naman Goyal, Peng-Jen Chen, and Francisco Guzmán. 2020. Findings of the WMT 2020 shared task on parallel corpus filtering and alignment. In *Proceedings of the Fifth Conference on Machine Translation*, pages 726–742, Online. Association for Computational Linguistics.
- Philipp Koehn, Hieu T. Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL*.
- Jun Lu, Xin Ge, Yangbin Shi, and Yuqi Zhang. 2020. Alibaba submission to the WMT20 parallel corpus filtering task. In *Proceedings of the Fifth Conference on Machine Translation*. Association for Computational Linguistics.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.
- Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.
- Javad Pourmostafa Roshan Sharami, Dimitar Shterionov, and Pieter Spronck. 2021. Selecting parallel in-domain sentences for neural machine translation using monolingual texts.
- Gowtham Ramesh, Sumanth Doddapaneni, Aravindh Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.
- Sukanta Sen, Mohammed Hasanuzzaman, Asif Ekbal, Pushpak Bhattacharyya, and Andy Way. 2021. Neural machine translation of low-resource languages using SMT phrase pair injection. *Natural Language Engineering*, 27(3):271–292.
- Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*.

Jörg Tiedemann. 2020. The Tatoeba Translation Challenge – Realistic data sets for low resource and multilingual MT. In *Proceedings of the Fifth Conference on Machine Translation*, pages 1174–1182, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*. Curran Associates, Inc.

A Models

We use a Transformer based architecture to train Hindi-Marathi and English-Marathi NMT models for all our experiments. The encoder of the Transformer consists of 6 encoder layers and 8 encoder attention heads. The encoder uses embeddings of dimension 512. The decoder of the Transformer also consists of 6 decoder layers and 8 decoder attention heads.

B Training Details

We use the OpenNMT-py library to train the Transformer based NMT models. The hyperparameter values are selected using manual tuning. The optimizer used was adam with betas (0.9, 0.98). The initial learning rate used was $5e-4$ with the inverse square root learning rate scheduler. We use 8000 warmup updates. The dropout probability value used was 0.1 and the criterion used was label smoothed cross entropy with label smoothing of 0.1. We use batch size of 4096 tokens. All the models were trained for 2,00,000 training steps.

We use Nvidia A100 GPUs with 40 GB memory to train our NMT models. The average training time of the models is 8 hours. The model parameters of each experiment are between 65M and 66M. The model parameters of the best models are given in the following section.

B.1 Baseline + LaBSE + PPI-LaBSE

In this experiment, the Hindi-Marathi and English-Marathi models consist of 66,263,552 (66.2M) and 65,812,992 (65.8M) parameters, respectively. We perform augmentation by providing the parallel corpus and extracted good quality parallel data to the OpenNMT-py tool as two corpora with the same weight (by default weight=1).

B.2 Baseline + LaBSE + PPI

In this experiment, the Hindi-Marathi and English-Marathi models consist of 66,348,544 (66.3M) and 66,554,368 (66.5M) parameters, respectively.