Data Selection as an Alternative to Quality Estimation in Self-Learning for Low Resource Neural Machine Translation

Idris Abdulmumin[1,2(✉)], Bashir Shehu Galadanci[2], Ibrahim Said Ahmad[2], and Rabiu Ibrahim Abdullahi[1]

[1] Ahmadu Bello University, Zaria, Kaduna, Nigeria
{iabdulmumin,raibrahim}@abu.edu.ng
[2] Bayero University, Kano, Kano, Nigeria
{bsgaladanci.se,isahmad.it}@buk.edu.ng

Abstract. For many languages, the lack of sufficient parallel data to train translation models have resulted in using the monolingual data, source and target, through self-learning and back-translation respectively. Most works that implemented the self-learning approach utilized a quality estimation system to ensure that the resulting additional training data is of sufficient quality to improve the model. However, the quality estimation system may not be available for many low resource languages, restricting the implementation of such approach to a very few. This work proposes the utilization of the data selection technique as an alternative to quality estimation. The approach will ensure that the models will learn only from the data that is closer to the domain of the test set, improving the performance of the translation models. While this approach is applicable to many, if not all, languages, we obtained similar and, in some implementations, even better results (+0.53 BLEU) than the self-training approach that was implemented using the quality estimation system on low resource IWSLT'14 English-German dataset. We also showed that the proposed approach can be used to improve the performance of the back-translation approach, gaining +1.79 and +0.23 over standard back-translation and self-learning with quality estimation enhanced back-translation respectively.

Keywords: Data selection · Self-learning · Quality estimation · Self-training · Neural machine translation · Machine translation

1 Introduction

Neural machine translation (NMT) [4,27,40] is the current state-of-the-art approach for building automatic translation systems [14,30]. Though it is simple,

This work is conducted as part of a PhD research supported by the NITDEF 2018.

© Springer Nature Switzerland AG 2021
O. Gervasi et al. (Eds.): ICCSA 2021, LNCS 12957, pp. 311–326, 2021.
https://doi.org/10.1007/978-3-030-87013-3_24

it out-performs other statistical machine translation approaches provided the parallel training data between the languages is sufficiently large [26,45]. The technique relies only on this huge parallel data to learn the probability of mapping sentences to the target language given a set of sentences in the source language [42]. This data is not only scarce between many languages but it is very expensive to create. This has influenced many research works that were aimed at investigating the possibility of applying other techniques and readily available resources to improve the performance of such translation models. These techniques include using other high resource languages through transfer learning [12,25,28,45] and the readily available large amounts of monolingual data through forward translation or self-training [2,37,43], dual learning [18] and back-translation [9,14,20,29,35].

The monolingual data in the source or target or both of the languages have been used in various studies to enhance the performance of models that are built for automatic machine translation, especially in settings that are considered low resource – where the training data is not sufficient to train models that are of acceptable standard. Currey et al. [11] proposed using a (moderately changed [8]) duplicate of the target sentences as the corresponding source sentences. Sennrich et al. [34] proposed using the back-translation of the target sentences that are generated with an intermediary reverse model while Ueffing [39] and Zhang and Zong [43] proposed using the forward translation approach to self-train the generating model. He et al. [18] used target and source language monolingual data through a dual learning approach and Gulcehre et al. [17] infused a language model that was trained on monolingual data to ensure that the decoder produce fluent translations.

The self-training (or self-learning) approach uses the translations of the monolingual source data to improve the performance of the generating model. The approach has been used in both phrase-based statistical [37,39] and neural machine translation systems [2,43]. The approach relies on the usually large amounts of readily available source language monolingual data to better translation models. The authentic parallel training data – which is scanty in low resource set-ups – is used to train a translation model. The additional pseudo-parallel training data is then used to generated by translating the available monolingual data using the model. The synthetic data is then paired with the available authentic data and the resulting large dataset is used to retrain an improved forward translation model. The synthetic data is often infested with mistakes. This is because the generating model is in itself not qualitative enough to generate error-free translations.

Earlier implementations of the approach, especially in [37,39], relied on a quality estimation system to ensure that the quality of the synthetic data that is generated is fit enough to improve the performance of the generating model. Although the work of Abdulmumin et al. [2] have shown that the generating model is capable of benefiting from the synthetic data even without estimating the quality, extensive studies by Abdulmumin et al. [3] (see implementation in Fig. 1) have found that models are better off when implemented with the

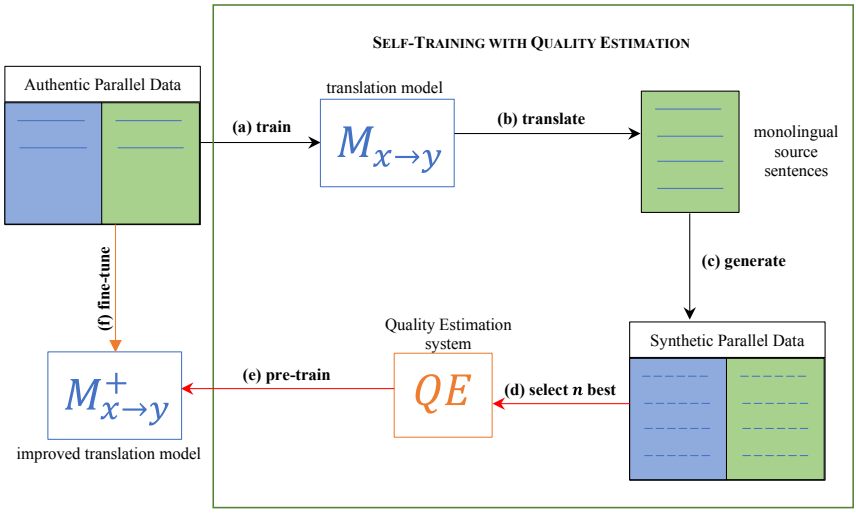


Fig. 1. Self-learning with quality estimation for improved machine translation models [3].

system. For most of low resource languages, the quality estimation system is not existent and so is the data for training such systems. This will, therefore, limit the application of the self-learning approach in such set-ups.

This work, therefore, proposes an alternative technique of generating a synthetic data that best benefits the generating model than the whole of the synthetic data. As with the principle of using the quality estimation to only select data that is considered best for the generating model, more (synthetic) data may not necessarily mean a better model [15]. The proposed approach utilizes a data selection algorithm to select a fraction of the monolingual data that is closest in domain to the source-side of the test set. This data is then translated and used as the additional data for the self-training approach. The approach is similar to the data selection described in Poncelas et al. [32] for back-translation but instead of comparing the domain of the synthetic data with that of the target-side of the test set (which can only be done after translating all of the available monolingual sentences), we compare the domain of the monolingual data with that of the source-side of the test set before translating. This will not only save translation time but it will also ensure that the domain match is as accurate as possible because all of the data used is authentic. Therefore, in this work, the following contributions are made:

– instead of relying on a quality estimation system that is only available for a few languages for implementing the more effective self-training approach, we propose a new alternative that relies on the more common data selection algorithms for selecting the more appropriate monolingual data,

– we applied data selection in self-learning in a way that ensures both the monolingual and source-side of the test data are natural, ensuring the domain match is as accurate as possible,
– we proposed a faster approach, ensuring that only a subset of the monolingual data is selected and translated instead of translating all of the available monolingual data and selecting only a subset for self-learning,
– we showed that the proposed technique is not only capable of producing comparable results with that of the self-training approach that is implemented with quality estimation but that in some implementations, it yielded better performances, and
– we successfully applied the proposed approach on low resource IWSLT 14 German- English data, improving over baseline and various implementations of the self-learning approaches.

2 Related Works

In this section, we present related works on self-training in automatic machine translation.

Ueffing [39] introduces self-learning in machine translation as an approach that utilizes the translations of a model for improving the performance of the generating model. In the proposed approach, sentences in the source language are translated and their quality is determined using a quality estimation system. The sentences whose quality is below a set minimum are discarded and then, the remaining generated parallel data is used to better the model.

Zhang and Zong [43] used the approach, naming it forward translation, in NMT for improving the encoder side of the model. The authors claimed that when the training data consists of an authentic source and synthetic target, only the model's encoder is improved because, they say, the impurities in the synthetic data will deteriorate the decoder's performance. To mitigate this, the authors freezes the parameters of the decoder when training on the synthetic data and only update them when the training instance is authentic.

Specia and Shah [37] propose an iterative version of [39]. Instead of discarding the low quality synthetic sentences, the monolingual source sentences whose translations are not selected during the self-learning approach are reserved and used in subsequent iterations to continue improving the model. The first iteration in this approach selects the top n translations using a quality estimation system for retrain the generating model. The rest of the iterations utilizes the same self-training approach but using the rest of the unselected monolingual sentences.

Abdulmumin et al. [2] investigated using the approach without data cleaning (quality estimation) to enhance the performance of an intermediary (backward) model in the back-translation approach. The proposed method was still found to benefit the model. But further studies in [3] has found that using quality estimation ensures a more effective self-learning approach.

The studies above showed benefits of using a quality estimation system to determine the synthetic data that is considered best for improving the performance of the generating model. The lack of such models can greatly affect the

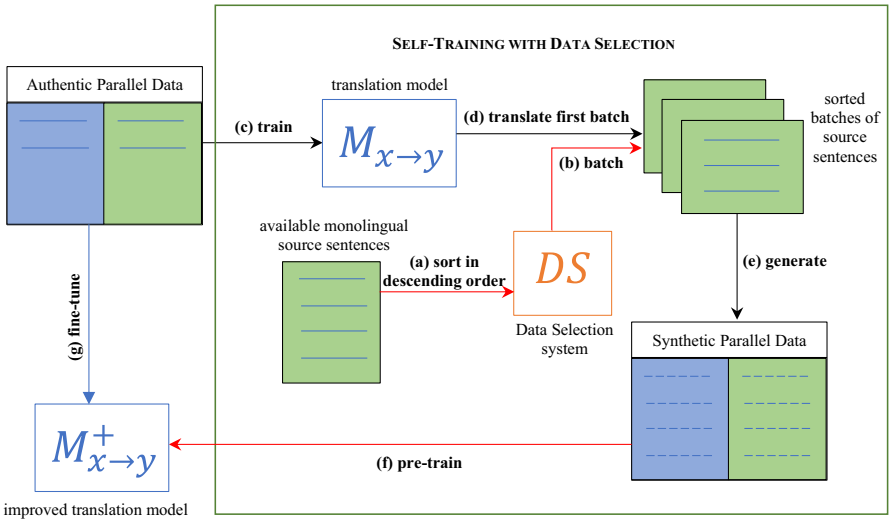


Fig. 2. Proposed self-learning with data selection for improved machine translation models.

rather efficient performance of such models. An alternative system that can produce similar or even better results is needed therefore in applying the self-learning approach especially in low resource languages.

3 Methodology

This section presents the proposed self-learning with data selection approach.

3.1 Overview of the Proposed Approach

The proposed approach is illustrated in Fig. 2 and presented in Algorithm 1. Our approach requires the monolingual source sentences, $X = \{(x^{(p)})\}_{p=1}^{P}$, the authentic parallel training data, $D = \{(x^{(q)}, y^{(q)})\}_{q=1}^{Q}$ and the test set, $T = \{(x^{(r)}, y^{(r)})\}_{r=1}^{R}$. Firstly, we used a data selection system to rank the monolingual source sentences according to how close their domain is to that of the source-side of the test set, $x^{(r)} \in T$. Afterwards, we selected the top a sentences from these sorted sentences. Secondly, a source-to-target model, $M_{x\to y}$, was trained on the authentic parallel data, D. Thirdly, the forward model was used to generate the synthetic parallel data, $D' = \{(x^{(a)}, y^{(a)})\}_{a=1}^{A}$, by translating the selected monolingual source data, X^a. Lastly, the resulting synthetic data is then used to enhance the performance of the forward model through pre-training and fine-tuning it on the synthetic and authentic data respectively. This method of training has been shown to best benefit any model that is trained on

Algorithm 1. Self-Learning with Data Selection

Input: Monolingual source data, $X = \{(x^{(p)})\}_{p=1}^{P}$, Authentic training data, $D = \{(x^{(q)}, y^{(q)})\}_{q=1}^{Q}$, and the Test set, $T = \{(x^{(r)}, y^{(r)})\}_{r=1}^{R}$.

procedure DATA SELECTION
    Using a Data Selection algorithm, sort X, from closest to farthest, based on how close the domain of $x^{(p)} \in X$ is to the domain of $x^{(r)} \in T$;
    Let X^a = the closest a monolingual source sentences;
end procedure
procedure SELF-LEARNING
    Train a forward model $M_{x \to y}$ on the authentic parallel data, D;
    Let $D' = \{(x^{(a)}, y^{(a)})\}_{a=1}^{A}$ = the synthetic data generated by $M_{x \to y}$, for all $x \in X^a$;
    Pretrain a model $M^{+}_{x \to y}$ on the synthetic parallel data, D';
    Finetune the forward model $M^{+}_{x \to y}$ on the authentic parallel data, D;
end procedure

Output: improved translation model, $M_{x \to y}$

synthetic and authentic data [2,3]. This approach is similar to the implementations of the self-learning approach in [3,39].

Alternatively, the self-learning with quality estimation was implemented in [37] to intelligently utilize all of the available data through an iterative process. As explained in Sect. 2, during the first iteration, the whole monolingual data is translated using a baseline model. Afterwards, a quality estimation model is used to select a subset (top n) of the monolingual data that is considered to be the most accurate. This data is then added to the training data to self-train an improved forward model. In the second iteration, the improved forward model is used to generate the synthetic data using all of the monolingual data whose translations were not previously selected and the best n of this batch are selected for the third iteration. This process is repeated until all of the monolingual data are used to train a better forward model. In this work, we implemented the same approach but instead of using the quality estimation system, we used the data selection algorithm to first sort and batch the monolingual data. The first batch was used for self-learning in the first iteration and subsequently, the remaining batches were used in each of the corresponding iterations to further improve the quality of the forward model.

We also implemented an iterative approach that uses only the closest (first) batch to continue to improve the forward model instead of all of the monolingual data. After self-training the forward model on the first batch, it was used to translate the same first batch and the process was repeated over some iterations. This was also implemented with the quality estimation system to compare the performances of both systems. The synthetic data that is estimated to be the best translated data was used in each of the iterations to further improve the model.

Table 1. The data used

Data	Train			Dev	Test
	Sentences	Words (vocab)			
IWSLT '14 German-English	153,348	En	De	6,970	6,750
		2,837,240 (50,045)	2,688,387 (103,796)		
WMT '14 German-English – Mono English	399,951	9,918,380 (266,640)		–	–

3.2 Data

The data used in this work is the IWSLT 2014 De-En (German-English) shared translation task [10]. We used the code provided by Ranzato et al. [33] for cleanup and splitting of the train, dev and test datasets. This resulted in 153,348, 6,970 and 6,750 parallel sentences respectively. The data statistics is shown in Table 1. For the additional monolingual data, we randomly selected and used 399,951 English monolingual sentences from the pre-processed [27] WMT 2014 German-English translation task [7]. For tokenization, we trained a byte pair encoding (BPE) [36] with 10,000 merge operations on the training dataset before applying it on the training, evaluation (dev) and test sets. For the self-training, a joint BPE is learned on the mixture of the synthetic and authentic parallel data. Afterwards, the training vocabulary was built on the mixed dataset.

3.3 Experimental Set-Up

In this work, a unidirectional Long Short-Term Memory (LSTM) [19], a recurrent neural network (RNN) machine translation (RNMT) architecture, with Luong attention [27] was used to built the encoder-decoder models. This architecture is summarized below. Our proposed method can be implemented using other architectures such as the Transformer [13,40] and the convolutional NMT (CNMT) [16,41]. The Neural Machine Translation (NMT), a sequence-to-sequence system, is made up of a neural encoder and another neural decoder for modeling the conditional probability of an output sentence given a sentence in the source language [4,27,38]. The encoder encodes the input sentence into a set of vectors while the decoder decodes these vectors and the previously predicted words into the desired output language, word by word, through an attention mechanism. Bahdanau et al. [4] proposed the attention mechanism for keeping track of context in longer sentences.

For building the models, we used the same settings as in [2]. This is the NMTSmallV1 configuration of the OpenNMT-tf [24] framework, which is implemented using the TensorFlow [1]. This consists of a 2-layer unidirectional LSTM encoder-decoder model with Luong attention [27]. Each of the encoder and decoder is made up of 512 hidden units. We used the Adam optimizer [23], a training batch size of 64 and a static learning rate of 0.0002. The models were

Table 2. Performances of the baseline and self-trained models with data selection and quality estimation for German-English NMT.

Checkpoint	Baseline	Self-training					
		+ *synth-all* [2]		+ *synth-qe* [3]		+ *synth-ds*	
		Pretrain	*Finetune*	*Pretrain*	*Finetune*	*Pretrain*	*Finetune*
Best (training step)	10.03 (65k)	6.02 (75k)	20.80 (125k)	6.24 (65k)	23.22 (165k)	7.35 (55k)	23.07 (140k)
Average	10.25	6.01	21.31	6.23	23.66	7.16	23.34

evaluated, using the bi-lingual evaluation understudy (BLEU) [31], on the dev set after each 5,000 steps. The training of each model was stopped when the maximum improvement observed is not more than 0.2 BLEU after evaluating four successive checkpoints. Unless stated otherwise, the training of all of the models was discontinued after the stopping condition is reached and for getting the optimal performance in most of the models, the average model is computed on the last 8 checkpoints. Since a joint BPE was used and the vocabulary was learned on the mixture of the two data, only the training data is changed during fine-tuning.

For quality estimation, we used the pre-trained Predictor-Estimator [22] QE model that was provided in the OpenKiwi's [21] quality estimation tutorial for synthetic German sentences. The model is based on an RNN architecture that, for estimating the quality of synthetic sentences, uses multi-level task learning. The model was trained using a neural predictor and neural quality estimation models that were jointly learnt using stack propagation [44]. For implementing the monolingual data selection, the frequency decay algorithm, FDA [5,6], was used. This algorithm uses the source-side of the test set as seed to rank, according to n-grams, the monolingual sentences due to their closeness to the domain of the test set.

4 Experiments and Results

Firstly, we used the data selection algorithm to sort the monolingual data from the closest to the farthest in their domain to the test set. The closest one-third of the monolingual sentences were selected for implementing the proposed self-learning with data selection approach. We then trained a baseline forward model (En-De) on the available parallel data for 80,000 steps. This model achieved the best single-checkpoint performance of 10.03 BLEU at the $65,000^{th}$ step. The best performance was obtained after averaging the last 8 checkpoints – 10.25 BLEU. This, taken as the baseline model, was used to generate the synthetic data for implementing the self-learning approaches. The performances of these improved forward models and the baseline are shown in Table 2.

Table 3. Performances of the iterative self-training with quality estimation as implemented in [37] and the same implementation of the self-learning approach but with data selection, utilizing all of the monolingual data for English-German NMT.

Method	Iteration	Performance of best checkpoint (training step)		Average (last 8 checkpoints)	
		Pretrain	Finetune	Pretrain	Finetune
SL + QE [3]	0	6.24 (65k)	23.22 (165k)	6.23	23.66
	1	16.18 (105k)	23.41 (200k)	15.88	23.62
	2	19.59 (155k)	23.69 (210k)	19.83	24.05
SL + DS	0	7.35 (55k)	23.07 (140k)	7.16	23.34
	1	20.42 (115k)	24.01 (170k)	20.54	24.24
	2	21.39 (150k)	24.03 (215k)	21.45	24.31

After the selected data was translated using the baseline forward model, the resulting synthetic data was labeled *synth-ds*. The baseline model was also used to generate *synth-all*, by translating all of the available monolingual data. The quality estimation system was used to estimate the accuracy of *synth-all*. The best one-third of this synthetic data was selected and labeled *synth-qe*. Each of *synth-ds* and *synth-qe* consisted of 133,295 parallel sentences, with 45,212 sentences found in both. Finally, the authentic parallel data was used with each of the three synthetic data were used to self-train improved forward models.

For the forward model that uses *synth-all*, it stopped training during the first phase – pre-training – at the $85{,}000^{th}$ training step, achieving the best single-checkpoint performance of 6.02 BLEU and an average checkpoint with 6.01 BLEU. Fine-tuning on the authentic parallel data resulted in an improvement of +14.78 and +15.30 BLEUs (20.80 and 21.31 BLEUs) for best and average checkpoints respectively. Altogether, the model was trained for 165,000 training steps. This model achieved a better performance than the baseline by 11.06 BLEU.

With *synth-qe*, the forward translation model achieved the best performance of 23.66 BLEU after averaging the last 8 checkpoints. This translates to +2.35 BLEU improvement over the forward model that was self-trained on *synth-all*. However, using *synth-ds* resulted in the overall best performance of 23.34 BLEU. This −0.32 BLEU less than that of the former model that was trained using *synth-qe*.

The performances of the iterative approaches with quality estimation and data selection are presented in Table 3. It can be observed that although the implementation using the quality estimation system outperformed the data selection at the self-learning stage, the later catches up and even outperformed the former at all stages of the iterations, achieving +0.62 and +0.26 respectively after the first and second iterations. The implementation with data selection even

Table 4. Performances of the iterative self-training with quality estimation and data selection approaches, utilizing only the best monolingual data for English-German NMT.

Method	Iteration	Performance of best checkpoint (training step)		Average (last 8 checkpoints)	
		Pretrain	Finetune	Pretrain	Finetune
+ QE	0	6.24 (65k)	23.22 (165k)	6.23	23.66
	1	18.71 (105k)	23.84 (160k)	18.82	24.10
	2	19.94 (110k)	23.79 (185k)	19.90	24.09
+ DS	0	7.35 (55k)	23.07 (140k)	7.16	23.34
	1	20.51 (95k)	24.05 (155k)	20.51	24.33
	2	21.72 (115k)	24.23 (185k)	21.87	24.62

Table 5. Time taken to estimate synthetic data quality when using all of the monolingual data and when using only the best monolingual data vs the time taken for sorting the monolingual data based on the domain closeness to the test set.

Methods	Iteration	Time taken
QE (all)	sl	31 h, 1 min and 29 s
	1	22 h, 54 min and 53 s
	2	–
QE (best)	sl	31 h, 1 min and 29 s
	1	30 h, 28 min and 4 s
	2	34 h, 42 min and 38 s
DS	–	0 h, 1 min and 7 s

pre-trained a forward model on the synthetic data that outperformed (+0.08 BLEU) the forward model that was self-trained on *synth-all* – see Tables 2 and 3.

Having selected from the monolingual data the closest sentences in domain to the test set, the same set was used in all of the iterations. The same subset of the monolingual data that was selected as *synth-qe* was also re-translated using the improved forward models in all of the iterations to compare. The performances of the resulting forward models are shown in Table 4. It can be observed also that the self-learning that was implemented with data selection achieved superior performance in all of the iterations, gaining +0.23 and +0.53 after the first and second iterations respectively.

Although all of the approaches produced similar performances, Table 5 shows that using data selection was faster especially during the data selection vs the quality estimation stage. On a computer with 2.30GHz frequency Intel CPU and 12GB RAM, the time it took to estimate the quality of the synthetic sentences when implementing the approach that utilizes all of the available synthetic data was 53 h, 56 min and 22 s (well over 2 days), while the approach that utilizes only

Table 6. Performances of the various forward models with and without the standard or the various enhanced back-translation approaches for German-English NMT. All of the methods were implemented using the pre-training and fine-tuning except the baseline model which was just indicated to compare performance.

Method	Performance of best checkpoint (training step)		Average (last 8 checkpoints)	
	Pretrain	Finetune	Pretrain	Finetune
Baseline [3]	–	20.30 (75k)	–	20.95
Standard BT [3]	5.43 (60k)	28.31 (155k)	5.13	28.83
SL + QE [3]	19.38 (160k)	30.01 (225k)	19.37	30.39
SL + DS	19.62 (95k)	30.21 (170k)	19.31	30.42
Iterative SL + QE (all mono) [3]	17.97 (120k)	29.55 (185k)	18.09	30.05
Iterative SL + DS (all mono)	19.21 (135k)	30.15 (190k)	19.31	30.62
Iterative SL + QE (best mono)	19.43 (115k)	30.12 (185k)	19.65	30.15
Iterative SL + DS (best mono)	19.58 (150k)	29.93 (215k)	19.63	30.10

the best synthetic data was 96 h, 12 min and 11 s (just over 4 days), compared to the 1 min and 7 s for sorting the data, a best case scenario of more than two days compared to the just over a minute in data selection.

4.1 Back-Translation

The various models that are trained were then taken as backward models and used to generate synthetic data for training German-English forward models to implement the back-translation approach. The performances of these models are shown in Table 6. It can be observed that both the quality estimation and the data selection enhanced self-learning influenced the improved performance in their respective forward models. The scores can be said to be statically similar, having a range of 0.57 BLEU. For the one-off self-learning with quality estimation and data selection, the difference was just a paltry 0.03 BLEU, almost the same difference as when we used only the best monolingual (synthetic) data for the self-training (0.05 BLEU). The huge difference was observed when we used all of the monolingual data for self-learning, a difference of 0.57 BLEU. However, all of the approaches above were significantly better than the baseline and the standard back-translation models.

5 Discussion

Neural machine translation models cannot be relied upon to generate acceptable translations between languages when the training data is not sufficiently large. These category of languages are referred to as low resourced. Using the synthetic data – generated from a large collection of monolingual sentences – through

self-learning has been investigated to improve the models' performances. The implementation of this approach has been shown to rely on quality estimation systems to better improve the models. But these quality estimation models are not available for many languages, especially those that are in the category of low resourced. Applying the self-learning approach without cleaning the synthetic data have also shown to be useful but not as effective. Also, the quality estimation systems expect that every sentence, no matter how large their number is, to be translated before estimating their quality and determining which are useful and which are not. This is not efficient.

This work, therefore, proposed the use of data selection to determine the sentences whose domain is closest to that of the test set to be used for self-learning. The approach has been used in back-translation to select from the synthetic data the sentences that share n-grams with the test set. Contrastingly, we select sentences from the authentic monolingual data before translation rather than from the synthetic sentences. The proposed approach achieved a performance that is similar but less than that of the current self-learning with quality estimation approach (−0.32 BLEU). But after the iterating the process, our proposed method reached the best performance of 24.31 BLEU, an improvement of +0.26 BLEU over the latter approach. We showed that even if the best of all of the available synthetic data was selected at each iteration, unlike in [37] where the best of the rest of the synthetic data is selected, implementing the iterative approach with data selection (using only the closest monolingual data) resulted in the best model – 24.10 vs 24.62 BLEUs respectively. The approach has not only shown to be just as effective (and even better), but that the number of monolingual sentences to be translated is the exact number that is needed for the self-learning approach. Determining the required sentences was also faster than estimating the quality of the synthetic data.

We also implemented the back-translation approach, taking all the previously trained models as backward models. The performances of the resulting forward models showed that the proposed approach can also be used to enhance the performance of the backward model in the back-translation method.

6 Conclusion and Future Work

In this work, we investigated using the monolingual data selection as an alternative to synthetic data quality estimation to improve self-learning in low resource neural machine translation. The approach was shown to be simpler and faster and can achieve a performance that is comparable or even better than using the quality estimation system. We showed, also, that the approach can be deployed in back-translation to enhance the performance of the backward model and that this improvement resulted in a better forward model.

For future work, we plan to study the efficacy of a hybrid of both quality estimation and data selection to further improve the performances of translation models. We plan, also, to study the applicability of the approach on rich-resourced languages.

References

1. Abadi, M., et al.: TensorFlow: a system for large-scale machine learning. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation, OSDI'16, pp. 265–283. USENIX Association, Berkeley (2016). http://dl.acm.org/citation.cfm?id=3026877.3026899
2. Abdulmumin, I., Galadanci, B.S., Isa, A.: Enhanced back-translation for low resource neural machine translation using self-training. In: ICTA 2020. CCIS, vol. 1350, pp. 355–371. Springer, Cham (2021). https://doi.org/10.1007/978-3-030-69143-1_28
3. Abdulmumin, I., Galadanci, B.S., Isa, A., Sinan, I.I.: A hybrid approach for improved low resource neural machine translation using monolingual data. arXiv:2011.07403 (2021)
4. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1409.0473
5. Biçici, E., Yuret, D.: Instance selection for machine translation using feature decay algorithms. In: Proceedings of the 6th Workshop on Statistical Machine Translation, pp. 272–283. Asian Federation of Natural Language Processing, Edinburg (2011). https://pdfs.semanticscholar.org/14c6/731b2839d40b0c45f447b07afce1cc996de4.pdf
6. Biçici, E., Yuret, D., Bicici, E., Yuret, D.: Optimizing instance selection for statistical machine translation with feature decay algorithms. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(2), 339–350 (2015). https://doi.org/10.1109/TASLP.2014.2381882
7. Bojar, O., et al.: Findings of the 2017 conference on machine translation (WMT17). In: Proceedings of the Second Conference on Machine Translation, Shared Task Papers, vol. 2, pp. 169–214. Association for Computational Linguistics, Copenhagen (2017). http://www.aclweb.org/anthology/W17-4717
8. Burlot, F., Yvon, F.F.: Using monolingual data in neural machine translation: a systematic study. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 144–155. Association for Computational Linguistics, Brussels (2018). https://doi.org/10.18653/v1/W18-6315. https://www.aclweb.org/anthology/W18-6315
9. Caswell, I., Chelba, C., Grangier, D.: Tagged back-translation. In: Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers), pp. 53–63. Association for Computational Linguistics, Florence (2019). https://doi.org/10.18653/v1/W19-5206. https://www.aclweb.org/anthology/W19-5206
10. Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Federico, M.: Report on the 11th IWSLT evaluation campaign, IWSLT 2014. In: Proceedings of the 11th Workshop on Spoken Language Translation, Lake Tahoe, CA, USA, pp. 2–16 (2014)
11. Currey, A., Miceli Barone, A.V., Heafield, K.: Copied monolingual data improves low-resource neural machine translation. In: Proceedings of the Second Conference on Machine Translation, vol. 1, pp. 148–156. Association for Computational Linguistics, Copenhagen (2017). https://doi.org/10.18653/v1/W17-4715
12. Dabre, R., et al.: NICT's supervised neural machine translation systems for the WMT19 news translation task. In: Proceedings of the Fourth Conference on Machine Translation (WMT), Florence, Italy, vol. 2, pp. 168–174 (2019)

13. Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., Kaiser, Ł.: Universal transformers. In: ICLR, pp. 1–23 (2019). http://arxiv.org/abs/1807.03819
14. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500. Association for Computational Linguistics, Stroudsburg (2018). https://doi.org/10.18653/v1/D18-1045. http://aclweb.org/anthology/D18-1045
15. Gascó, G., Rocha, M.A., Sanchis-Trilles, G., Andrés-Ferrer, J., Casacuberta, F.: Does more data always yield better translations? In: EACL 2012–13th Conference of the European Chapter of the Association for Computational Linguistics, Proceedings, pp. 152–161 (2012)
16. Gehring, J., Michael, A., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. In: Precup, D., Teh, Y.W. (eds.) Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1243–1252. PMLR, Sydney (2017). http://proceedings.mlr.press/v70/gehring17a.html
17. Gulcehre, C., Firat, O., Xu, K., Cho, K., Bengio, Y.: On integrating a language model into neural machine translation. Comput. Speech Lang. **45**(2017), 137–148 (2017). https://doi.org/10.1016/j.csl.2017.01.014
18. He, D., et al.: Dual learning for machine translation. In: Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, pp. 820–828. Curran Associates Inc., USA (2016). http://dl.acm.org/citation.cfm?id=3157096.3157188
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
20. Imamura, K., Fujita, A., Sumita, E.: Enhancement of encoder and attention using target monolingual corpora in neural machine translation. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation, pp. 55–63. Association for Computational Linguistics, Melbourne (2018)
21. Kepler, F., Trénous, J., Treviso, M., Vera, M., Martins, A.F.T.: OpenKiwi: an open source framework for quality estimation. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 117–122. Association for Computational Linguistics, Florence (2019)
22. Kim, H., Lee, J.H., Na, S.H.: Predictor-estimator using multilevel task learning with stack propagation for neural quality estimation. In: Proceedings of the Conference on Machine Translation (WMT), vol. 2, pp. 562–568. Association for Computational Linguistics, Copenhagen (2017). https://doi.org/10.18653/v1/w17-4763
23. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings, San Diego, CA, USA (2015). http://arxiv.org/abs/1412.6980
24. Klein, G., Kim, Y., Deng, Y., Senellart, J., Rush, A.: OpenNMT: open-source toolkit for neural machine translation. In: Proceedings of ACL 2017, System Demonstrations, pp. 67–72. Association for Computational Linguistics, Vancouver (2017). https://www.aclweb.org/anthology/P17-4012
25. Kocmi, T., Bojar, O.: CUNI submission for low-resource languages in WMT news 2019. In: Proceedings of the Fourth Conference on Machine Translation (WMT), Florence, Italy, vol. 2, pp. 234–240 (2019)
26. Koehn, P., Knowles, R.: Six challenges for neural machine translation. In: Proceedings of the First Workshop on Neural Machine Translation, pp. 28–39. Association for Computational Linguistics, Vancouver (2017). https://doi.org/10.18653/v1/w17-3204

27. Luong, T., Pham, H., Manning, C.D.: Effective approaches to attention-based neural machine translation. In: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 1412–1421. Association for Computational Linguistics, Stroudsburg (2015). https://doi.org/10.18653/v1/D15-1166. http://aclweb.org/anthology/D15-1166
28. Nguyen, T.Q., Chiang, D.: Transfer learning across low-resource, related languages for neural machine translation. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing, vol. 2, pp. 296–301. Asian Federation of Natural Language Processing (2017)
29. Niu, X., Denkowski, M., Carpuat, M.: Bi-directional neural machine translation with synthetic parallel data. In: Proceedings of the 2nd Workshop on Neural Machine Translation and Generation. pp. 84–91. Association for Computational Linguistics, Melbourne (2018). https://doi.org/10.18653/v1/W18-2710. https://www.aclweb.org/anthology/W18-2710
30. Ott, M., Edunov, S., Grangier, D., Auli, M.: Scaling neural machine translation. In: Proceedings of the Third Conference on Machine Translation: Research Papers, pp. 1–9. Association for Computational Linguistics, Stroudsburg (2018). https://doi.org/10.18653/v1/W18-6301. http://aclweb.org/anthology/W18-6301
31. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL '02, pp. 311–318. Association for Computational Linguistics, Stroudsburg (2002). https://doi.org/10.3115/1073083.1073135
32. Poncelas, A., Maillette de Buy, G.W., Way, A.: Adaptation of machine translation models with back-translated data using transductive data selection methods. CoRR abs/1906.0 (2019). http://arxiv.org/abs/1906.07808
33. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. In: International Conference on Learning Representations (2016)
34. Sennrich, R., Haddow, B., Birch, A.: Edinburgh neural machine translation systems for WMT 16. In: Proceedings of the First Conference on Machine Translation, Shared Task Papers, vol. 2, pp. 371–376. Association for Computational Linguistics, Stroudsburg (2016). https://doi.org/10.18653/v1/W16-2323, http://aclweb.org/anthology/W16-2323
35. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 86–96. Association for Computational Linguistics, Berlin (2016)
36. Sennrich, R., Haddow, B., Birch, A.: 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), pp. 1715–1725. Association for Computational Linguistics, Stroudsburg (2016). https://doi.org/10.18653/v1/P16-1162. http://aclweb.org/anthology/P16-1162
37. Specia, L., Shah, K.: Machine translation quality estimation: applications and future perspectives. In: Moorkens, J., Castilho, S., Gaspari, F., Doherty, S. (eds.) Translation Quality Assessment. MTTA, vol. 1, pp. 201–235. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-91241-7_10
38. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: NIPS (2014)

39. Ueffing, N.: Using monolingual source-language data to improve MT performance. In: International Workshop on Spoken Language Translation, Kyoto, Japan, pp. 174–181 (2006)
40. Vaswani, A., et al.: Attention is all you need. In: 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA (2017)
41. Wu, F., Fan, A., Baevski, A., Dauphin, Y.N., Auli, M.: Pay less attention with lightweight and dynamic convolutions. arXiv:1901.10430v2, pp. 1–14 (2019)
42. Yang, Z., Chen, W., Wang, F., Xu, B.: Effectively training neural machine translation models with monolingual data. Neurocomputing **333**, 240–247 (2019). https://doi.org/10.1016/j.neucom.2018.12.032
43. Zhang, J., Zong, C.: Exploiting source-side monolingual data in neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1535–1545. Association for Computational Linguistics, Austin (2016). https://doi.org/10.18653/v1/d16-1160
44. Zhang, Y., Weiss, D.: Stackpropagation: improved representation learning for syntax. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1557–1566. Association for Computational Linguistics, Berlin (2016). http://www.aclweb.org/anthology/P16-1147
45. Zoph, B., Yuret, D., May, J., Knight, K.: Transfer learning for low-resource neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1568–1575. Association for Computational Linguistics, Austin (2016). https://doi.org/10.18653/v1/D16-1163. https://www.aclweb.org/anthology/D16-1163