HintedBT: Augmenting Back-Translation with Quality and Transliteration Hints

Sahana Ramnath, Melvin Johnson,* Abhirut Gupta,* Aravindan Raghuveer
Google Research
{sahanaramnath, melvinp, abhirut, araghuveer}@google.com

Abstract

Back-translation (BT) of target monolingual corpora is a widely used data augmentation strategy for neural machine translation (NMT), especially for low-resource language pairs. To improve effectiveness of the available BT data, we introduce HintedBT—a family of techniques which provides hints (through tags) to the encoder and decoder. First, we propose a novel method of using *both high and low quality* BT data by providing hints (as source tags on the encoder) to the model about the quality of each source-target pair. We don't filter out low quality data but instead show that these hints enable the model to learn effectively from noisy data. Second, we address the problem of predicting whether a source token needs to be translated or transliterated to the target language, which is common in cross-script translation tasks (i.e., where source and target do not share the written script). For such cases, we propose training the model with additional hints (as target tags on the decoder) that provide information about the *operation* required on the source (translation or both translation and transliteration). We conduct experiments and detailed analyses on standard WMT benchmarks for three cross-script low/medium-resource language pairs: {Hindi,Gujarati,Tamil}→English. Our methods compare favorably with five strong and well established baselines. We show that using these hints, both separately and together, significantly improves translation quality and leads to state-of-the-art performance in all three language pairs in corresponding bilingual settings.

1 Introduction

Neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Bahdanau et al., 2014; Sutskever et al., 2014; Wu et al., 2016; Hassan et al., 2018) models have become the state-of-the-art approach to machine translation. However, NMT models are data hungry and have been shown to under-perform in low-resource scenarios (Koehn and Knowles, 2017). Various supervised and unsupervised techniques (Song et al., 2019; Gulcehre et al., 2015) have been proposed to address the paucity of high-quality parallel data in such cases. *Back-translation* (Sennrich et al., 2016b) is one such widely used data augmentation technique in which synthetic parallel data is created by translating monolingual data in the target language to the source language using a baseline system. However, in order to get high quality parallel back-translated (BT) data, we need a high quality target→source translation model (Burlot and Yvon, 2019). This in turn depends on having a substantial amount of high quality parallel (bitext) data already available. For low-resource languages, both the quantity and quality of bitext data is limited, leading to poor back-translation models. Existing methods either use all BT data available (Sennrich and Zhang, 2019), or use various cleaning techniques to identify and filter out lower quality BT data (Khatri and Bhattacharyya, 2020; Imankulova et al., 2017). However, filtering reduces the amount of data available for training in a scenario which is already low-resource. How to efficiently use back-translation data in a situation where data is both scarce and of varied quality is the first key challenge we tackle in this paper.

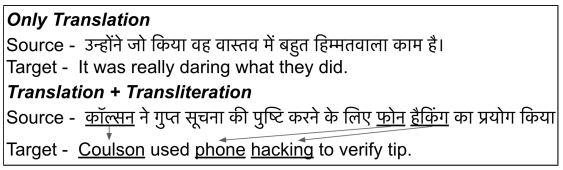

Figure 1: Examples from the WMT 2014 Hindi→English test set. The top example is a case of only translation, and the bottom one is a case where some words in the source (a named-entity "Coulson", and English words 'phone', 'hacking' written in Hindi) need to be transliterated.

The second challenge that arises increasingly

*equal contribution

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 1717–1733
November 7–11, 2021.

often in low-resource MT is that of cross-script NMT: translation tasks where the source and target languages do not share the same script. Cross-script NMT tasks have been steadily increasing in the WMT shared news translation tasks[1] over the past few years (28% of tasks in 2017 and 2018, 44% in 2019, and 63% in 2020). Cross-script NMT models must implicitly predict whether a source token needs to be translated or transliterated (see example in Figure 1). Lack of shared vocabulary coupled with low data quantity and quality makes cross-script NMT in low-resource settings a very challenging task.

In this work, we propose **HintedBT**, a family of techniques that provide hints to the model to make the limited BT data even more effective. We present results on three cross-script WMT datasets: Hindi(hi)/Gujarati(gu)/Tamil(ta)→English(en). In our first proposed HintedBT method, ***Quality Tagging***, we use tags to provide hints to the model about the quality of each source-target BT pair. In the second method, ***Translit Tagging***, we use tags to address the cross-script NMT challenge described above: we force the decoder to predict the *operation* that needs to be done on the source - only translation (or) both translation + transliteration, in addition to predicting the translated sentence. The correct operation is provided as an additional tag during training.

We make the following contributions in this paper:

1. Two novel hinting techniques: Quality Tagging (Section 3) and Translit Tagging (Section 4) to address two key challenges in low-resource cross-script MT.
2. Extensive experiments and comparisons to competitive baselines which show that a combination of our methods outperform bilingual state-of-the-art models for all three languages studied (Section 5, 6). Table 1 shows BLEU scores of our methods compared to SoTA.
3. Applications of proposed techniques in other situations that arise commonly in low-resource language settings (Section 7).

Lang. Pair	SoTA	This work
hi→en	16.7 [Bilingual] (Matthews et al., 2014)	32.0
	28.7 [Multilingual] (Wang et al., 2020)	
gu→en	18.4 [Bilingual] (Bei et al., 2019)	20.8
	24.9 [Multilingual] (Li et al., 2019)	
ta→en	15.8 [Bilingual] (Parthasarathy et al., 2020)	17.2
	21.5 [Multilingual] (Chen et al., 2020)	

Table 1: Current SoTA versus our contributions. Our methods beat the bilingual SoTA for all three language pairs, and are competent with the multilingual SoTA, despite not using additional information in the form of pivot languages and/or multilingual models.

2 Related Work

Leveraging monolingual data for NMT: Initial efforts in this space focused on using target-side language models (He et al., 2016; Gulcehre et al., 2015). Recently, *back-translation*, first introduced for phrase-based models (Bertoldi and Federico, 2009; Bojar and Tamchyna, 2011) and popularized for NMT by Sennrich et al. (2016b), has been widely used. It has been shown that the quality of the back-translated data matters (Hoang et al., 2018; Burlot and Yvon, 2018). Given this finding, several works have performed filtering using sentence-level similarity metrics on the round-trip translated target and the original target (Imankulova et al., 2017; Khatri and Bhattacharyya, 2020), or cross-entropy scores (Junczys-Dowmunt, 2018). Several works have looked into iterative back-translation for supervised and unsupervised MT (Hoang et al., 2018; Cotterell and Kreutzer, 2018; Niu et al., 2018; Lample et al., 2018; Artetxe et al., 2018).

Multilingual models: Another direction in low-data settings is to leverage parallel data from other language-pairs through pre-training or jointly training multilingual models (Zoph et al., 2016; Johnson et al., 2017; Nguyen and Chiang, 2017; Gu et al., 2018; Kocmi and Bojar, 2018; Aharoni et al., 2019; Arivazhagan et al., 2019). Amongst recent WMT submissions, Chen et al. (2020); Zhang et al. (2020b); Kocmi and Bojar (2019) train multilingual models for ta→en and gu→en, whereas Goyal and Sharma (2019); Bawden et al. (2019); Dabre et al. (2019); Li et al. (2019) pivot through Hindi, or transliterate Hindi data to Gujarati for training gu→en models. Wang et al. (2020) train a multilingual model for hi→en with a multi-task learning framework that jointly trains the model on a translation task on parallel data and two denoising tasks on monolingual data. Improving low-resource MT without leveraging data from other language-pairs

[1] http://www.statmt.org/wmt20/translation-task.html

has received lesser attention, notably in Nguyen and Chiang (2018); Ramesh and Sankaranarayanan (2018); Sennrich and Zhang (2019). In this work, we experiment with bilingual models only, using no additional information from other language pairs.

Using tags during NMT training: Tags on the source side of NMT systems have been used to denote the target language in a multilingual system (Johnson et al., 2017), formality or politeness (Yamagishi et al., 2016; Sennrich et al., 2016a), gender information (Kuczmarski and Johnson, 2018), the source domain (Kobus et al., 2017), translationese (Riley et al., 2020), or whether the source is a back-translation (Caswell et al., 2019a). In this work, we use tags on the source side to represent the quality of the BT pair, and tags on the target side to represent the operation done on the source (translation, or translation + transliteration).

3 Quality-based Tagging of the BT data

In low-resource scenarios, where bitext data is low in quantity and quality, BT data will likely contain pairs with varying quality. So far, there have been two broad approaches to deal with BT data: (a) full-BT: use all the BT data without considering the quality of the BT pairs (Sennrich and Zhang, 2019) (b) topk-BT: use only high quality BT pairs by introducing some notion of quality between the source and target (Khatri and Bhattacharyya, 2020; Imankulova et al., 2017). The full-BT method suffers from the disadvantage that it mixes the good and bad quality data, hence confusing the model. This was one of the primary motivations for introducing topk-BT models. However topk-BT models, while being *quality-aware*, filter away a substantial chunk of the parallel data which could be harmful in low-resource settings.

In this work, we introduce a third type of using BT data called ***Quality Tagging***. This approach uses *all* the BT data by utilizing quality information about each instance. Our method extends the *Tagged BT* approach (Caswell et al., 2019a) that uses "tags" or markers on the source to differentiate between bitext and BT data. We attach multiple tags to the BT data, where each tag corresponds to a *quality bin*. The quality bin provides a hint of the quality of the BT pair being tagged. We use LaBSE (Feng et al., 2020), a BERT-based language-agnostic cross-lingual model to compute sentence embeddings. The cosine similarity between these source and target embeddings is treated as the quality score of the BT pair. BT pairs are then binned into k groups based on the quality score, and the bin-id is used as a tag in the source while training (cf. examples in Figure 2).

Quality Tagging	
<bin4> आप नंबर प्लेट चाहते हैं?	You want the number plate?
<bin1> कभी -कभी वह स्क्रीन पर आती है और ताकती है।	Sometimes she turns and stares at the screen.
Translit Tagging	
हम सब एक -दूसरे का समर्थन करते हैं।	<Txn> We're all supportive of each other.
आप नंबर प्लेट चाहते हैं?	<Both> You want the number plate?
Quality Tagging + Translit Tagging	
<bin4> आप नंबर प्लेट चाहते हैं?	<Both> You want the number plate?

Figure 2: Quality tags are prepended to the source, with <bin1>/<bin4> samples being the lowest/highest quality respectively. Translit tags are prepended to the target, with <Txn>/<Both> being translation only or translation + transliteration respectively. Correct translation of <bin1> example: *Sometimes she comes on the screen and stares.*

We explore three design choices in Quality Tagging below: a) Bin Assignment: How to assign a particular BT pair to a bin? b) Number of bins to use c) Bitext Quality Tagging.

Design Choice 1 - Bin Assignment: We have two direct options: *Equal Width Binning* or *Equal Volume Binning*. In *Equal Width Binning*, we divide quality score range into k intervals of equal size. Each interval then corresponds to a bin and each BT pair is assigned to the bin which contains its quality score. In *Equal Volume Binning* we sort the N data points by their quality score and divide points into k equally sized groups. Each group then corresponds to a bin. We see that Equal Width Binning (and other size-agnostic approaches like k-means) can cause severely size-unbalanced bins, with the lowest bin(s) not adding any signal at all. This is primarily because the cosine similarity used as quality score is language-pair agnostic and not calibrated to well separated quality bins. Equal Volume binning addresses this concern while also providing sufficiently inherent quality-based clusters with a good choice of k.

Design Choice 2 - Number of Bins: We experimented with different number of bins (see detailed results in Appendix E). From the dev-BLEU scores, we found that for hi→en and gu→en, four bins provide the best performance, while for ta→en either three or four bins work equally well. We uniformly use four bins for the sake of simplicity and point out that deeper analysis of the interplay between bitext, BT quality and number of bins is an interesting area of future work.

Design Choice 3 - Bitext Quality Tagging: We have three choices for this question: a) Bitext is left untagged. b)Bitext is always tagged with the highest quality bin. c) A Bitext pair is also scored using LaBSE and assigned to a bin just as a BT pair would be. We discuss this design choice further in Section 5.5.

4 Translit Tagging of the BT data

When the source and target are written in different scripts, certain words in the source explicitly need to be transliterated to the target language, such as entities, or target language words written in the source script (see example in Figure 1). In such cases, the model needs to identify which source words should be translated to the target language, and which need to be transliterated. To understand the prevalence of this pattern, we split the test data into two categories: *{Txn, Both}*. *'Txn'* means the target sentence requires translating every source word and *'Both'* means the a mix of translation and transliteration is needed to generate the target from the source words. Then we compare the percentage of sentence pairs in each category for the hi/gu/ta→en WMT test sets. For each word in the source sentence, we use FST transliteration models (Hellsten et al., 2017) to generate 10 English (i.e., the target language) transliterations. If any of these transliterations are present in the corresponding target, we categorize the pair as *Both*, else as *Txn*. From Table 3, we see that for all the three WMT test sets, ∼60-80% of the test corpora require a mix of translation and transliteration to be performed on the source sentences. Further details about the FST models are included in Appendix D.

To utilize this information about cross-script data in training, we propose a novel method: *Translit Tagging*. We use the aforementioned methodology to split the train data into two categories: *{Txn, Both}* as before. We then convert this information into tags, which we prepend to the *target* sentence (refer Figure 2 for an example). This method teaches the model to predict if the transliteration operation is required or not for the given source sentence, hence the name 'translit' tagging. During inference, the model first produces the translit tag on the output, before producing the rest of the translated text. Another option is to present translit tags on the source side while training. This method does not perform as well and also has practical challenges that we describe in detail in Appendix F.

5 Experiments

5.1 Datasets

Table 2 describes the train, dev, and test data used in our experiments. We train source→target and target→source NMT models on the available bitext data for all language pairs. We use the latter to generate synthetic back-translation data from the WMT Newscrawl 2013 English monolingual corpus.

5.2 Model Architecture

We train standard Transformer encoder-decoder models as described in Vaswani et al. (2017). The dimension of transformer layers, token embeddings and positional embeddings is 1024, the feedforward layer dimension is 8192, and number of attention heads is 16. We use 6 layers in both encoder and decoder for the hi→en models and 4 layers for the gu→en and ta→en models. For training, we use the Adafactor optimizer with $\beta_1 = 0.9$ and $\beta_2 = 0.98$, and the learning rate is varied with warmup for 40,000 steps followed by decay as in Vaswani et al. (2017). We perform all experiments on TPUs, and train models for 300k steps. We use a batch size of 3k across all models and tokenize the source and target using WordPiece tokenization (Schuster and Nakajima, 2012; Wu et al., 2016). Further details on hyper-parameter selection and experimental setup can be found in Appendix B.

5.3 Evaluation Metrics

We use SacreBLEU[2] (Post, 2018) to evaluate our models. For human evaluation of our data, we ask raters to evaluate each source-target pair on a scale of 0-6 similar to Wu et al. (2016), where 0 is the lowest and 6 is the highest (more details in Appendix C).

5.4 Baselines

We present the following five baseline models to compare our methods against. Baselines 3-5 are our re-implementations of relevant prior work which introduce different methods of improving on the full-BT baseline (Baseline 2).

1. **bitext** - Model trained only on bitext data. The size of train data is shown in Table 5.
2. **bitext + full-BT** - Model trained on bitext data and an additional 23M back-translated pairs.

[2]SacreBLEU Hash: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.0

	Train	WMT newsdev/test
hi→en	IIT Bombay en-hi Corpus (Kunchukuttan et al., 2017) - 1.68M pairs	WMT-2014 (520/2.5k pairs)
gu→en	WMT-2019 gu-en, TED2020 (Reimers and Gurevych, 2020), GNOME & Ubuntu (Tiedemann, 2012), OPUS (Zhang et al., 2020a) - 162k pairs	WMT-2019 (3.4k/1k pairs)
ta→en	WMT-2020 ta-en, GNOME (Tiedemann, 2012), OPUS (Zhang et al., 2020a) - 630k pairs	WMT-2020 (2k/1k pairs)

Table 2: Datasets used for training. The dev and tests are used from the WMT corpus.

WMT Test Set	Txn: Translation Only	Both: Translation + Transliteration
hi→en (2014)	21.3%	78.7%
gu→en (2019)	30.6%	69.4%
ta→en (2020)	40.5%	59.5%

Table 3: % of the WMT test sets where task is either only translation or a mix of translation & transliteration

3. **bitext + Iterative-BT** - Iterative training of models in the forward and reverse directions (Hoang et al., 2018). In our experiments, models are trained with two iterations of back-translation. We also study the interaction of Iterative-BT with HintedBT in Section 5.8.
4. **bitext + tagged-full-BT** - Model trained on bitext data and tagged full-BT data (Caswell et al., 2019b). A tag is added to the source in every BT pair to help the model distinguish between natural (bitext) and synthetic (BT) data.
5. **bitext + LaBSE topk-BT** - Model trained on bitext data and *topk* best quality BT pairs. Quality is estimated using LaBSE scores, and we grid-search with at least 6 LaBSE threshold values and choose the one which gives the best BLEU on the dev set (see Appendix A for more details). The chosen threshold yields 20M BT sentences for hi→en, 10M for gu→en and 5M for ta→en.

We report the performance of these baseline models on the WMT test sets in rows 1-5 in Table 4. Adding BT data alone (Row-2) provides a significant improvement in performance for hi→en (+58%) and gu→en (+78%) over the plain bitext baseline (Row-1). However for ta→en, the improvement is comparatively smaller (+24.7%). To understand this deviation further, we conduct a human evaluation (Section 5.3) on a random 500 samples of the bitext data. The results are reported in Table 5. We see that the ta→en bitext data is much poorer in quality compared to the other two pairs. This affects the quality of the back-translation model and hence influences the results of a few more experiments we report further.

Next we see that iterative back-translation (Row-3) and tagged back-translation (Row-4) do improve the performance for gu→en, ta→en but not for hi→en, when compared to Row-2. Comparison between full-BT (Row-2) and topk-BT (Row-5) shows choosing high quality BT data instead of using all the BT data proves beneficial for all 3 language pairs.

5.5 Quality Tagging

As explained in Section 3, we assign each BT pair to one of four quality bins that have equal volume of pairs in them. Table 6 presents the mean quality score as annotated by humans for different bins. We see a perceptible difference in the quality of data across bins for all languages. This confirms our hypothesis that BT data will be of varied quality. It reinforces faith in our choice of four equal volume bins and also in LaBSE as a method for automatic quality evaluation.

We now explore the choice of how to tag the bitext data (i.e., design choice 3), using human evaluation of the bitext and BT data (see Table 7). To re-iterate, we perform human evaluation of data by having raters evaluate each source-target pair on a scale of 0-6, with 0 being the lowest, and 6 being the highest (more details in Appendix C). For hi→en, both bitext and BT data are of high quality (>4). Hence, we decide to tag the bitext with the highest quality bin <bin4>. For gu→en, the BT data is of lower quality compared to the bitext. Hence, we decide to leave the bitext untagged, making the BT data's quality tags both an indicator of quality, as well as an indicator that the data is synthetic. For ta→en, both bitext and BT are of lower quality (<4), with the bitext's quality being slightly higher. Hence, here as well, we decide to leave the bitext untagged. We further demonstrate our choices using experiments. From Table 7, we see that for hi→en, tagging with <bin4> works best, while for gu→en leaving it untagged works best. For ta→en, there is no clear winner.

For the remaining experiments in this paper, we stick to this assignment for bitext tagging: gu→en and ta→en (untagged), hi→en (<bin4> tag).

	#	Modeling Methodology	hi→en	gu→en	ta→en
WMT Data Baselines	1	bitext	19.5	8.4	11.3
	2	bitext + full-BT	30.9	15.0	14.1
Prior Work	3	bitext + Iter-BT	29.2	16.5	14.9
	4	bitext + tagged-full-BT	30.2	17.0	16.0
	5	bitext + *LaBSE* topk-BT	31.2	16.0	16.4
HintedBT	6	bitext + full-BT + *LaBSE* quality tags	31.2	17.6	15.5
	7	bitext + full-BT + translit-tags	31.0	15.2	15.0
	8	bitext + full-BT + *LaBSE* quality tags + translit-tags	**31.6**	17.9	16.0
Iterative HintedBT	9	bitext + tagged-Iter-BT	30.0	20.5	16.5
	10	bitext + Iter-BT + *LaBSE* quality tags	29.9	20.0	**17.2**
	11	bitext + Iter-BT + *LaBSE* quality tags + translit-tags	29.5	**20.8**	16.3

Table 4: Performance of models on WMT test sets.

Data Quality	hi→en	gu→en	ta→en
bitext	4.16±0.15	4.37±0.15	3.73±0.17
full-BT	4.41±0.11	3.42±0.14	3.51±0.13

Table 5: Mean human quality scores on 500 samples of bitext and full-BT data alone with 95% CI. Integer ratings for individual sentence pairs lie in [0,6].

	Bin 1	Bin 2	Bin 3	Bin 4
hi→en	4.05 ±0.13	4.47 ±0.10	4.53 ±0.10	4.87 ±0.09
gu→en	2.28 ±0.17	2.79 ±0.15	3.18 ±0.15	3.85 ±0.15
ta→en	1.44 ±0.17	2.78 ±0.15	3.31 ±0.14	3.99 ±0.13

Table 6: Mean human quality scores for the 4 quality bins along with 95% CIs.

We present results of the quality tagged models in Row-6 of Table 4. First when we compare Row-6 with full-BT in Row-2, we see that quality tagging always yields higher BLEU. Same pattern exists with Row-3 where quality tagging always outperforms Iterative-BT for all language pairs. This is an important result because, while Iterative-BT is effective, it is also very computationally expensive. Quality tagging is able to produce better results than Iterative-BT with far lesser computational costs. Quality Tagging again outperforms both tagged-BT and topk-BT for hi→en and gu→en. For ta→en, topk-BT still has the best BLEU. We delve into more details on why this happens in Section 6.1. To summarize, we see quality tagging provides the best performance across all previous baselines (except in two ta→en instances). In addition, quality tagging is far more efficient than topk-BT in terms of computational resources since topk-BT requires multiple models to be trained for the threshold parameter search.

5.6 Translit Tagging

As explained in Section 4, we train the decoder to generate the translit tag (*'Txn'* or *'Both'*) along with the target sentence. During evaluation, we remove the translit tag which the model has produced in the output. Row-7 in Table 4 shows the BLEU of the translit tagging models, and the corresponding baseline is Row-2. As we can see, translit tagging improves the performance of all three language-pairs over the baseline.

Bitext tagging	hi→en	gu→en	ta→en
Untagged	30.0	17.6	15.5
Tagged with <bin4>	31.2	16.8	15.6
LaBSE Quality Tags	30.9	16.2	15.7

Table 7: Quality tagging on full-BT data, with bitext tagged/untagged

5.7 HintedBT: Quality + Translit Tagging

We combine our methods of *Quality Tagging* and *Translit Tagging* in this experiment: we tag the source with quality tags (as per Section 5.5), and we tag the target with translit-tags (as per Section 5.6). We report the results as Row-8 in Table 4. We see that for all 3 language pairs, the combination of these 2 methods outperforms both methods individually (comparing with Rows 6 and 7). For hi→en, this combination gives the overall best results of **31.6**, and to the best of our knowledge, this outperforms the bilingual SoTA (Matthews et al., 2014) as well as the multilingual SoTA (Wang et al., 2020) for hi→en. For gu→en, the combination produces +1.9 over an already strong topk-BT baseline. However for ta→en, topk-BT still remains as the best method thus far.

5.8 Iterative HintedBT

In this section, we apply Iterative Back-Translation (Hoang et al., 2018) in combination with the two methods in HintedBT: Quality Tagging and Translit Tagging. The goal here is to understand if our method is able to capitalize on the gains of Iterative-

BT or whether its gains are subsumed by a powerful method like Iterative-BT. We run Iterative-BT first with quality tagging alone, and next Iterative-BT with the combination of both quality tagging and translit tagging. As an additional baseline, we also run Iterative-BT with back-translation tagging as in Row-4 (Caswell et al., 2019b). We run two iterations of back-translation in all experiments.

We perform quality tagging for models in both directions using the Equal Volume method with four bins. In every round, we generate the BT, compute the LaBSE scores and assign each pair to the right bin and train the model. Row-10 in Table 4 shows BLEU when quality tagging is applied along with Iterative-BT. Comparing Row-10 with its corresponding full-BT baseline in Row-6, we see that the iterative version performs even better, with gu→en and ta→en getting BLEU scores of **20.0** and **17.2**, respectively. To the best of our knowledge, this outperforms the bilingual SoTA for ta→en (Parthasarathy et al., 2020).

Row-11 shows the performance when Iterative-BT is combined with both Quality Tagging and Translit Tagging. Comparing Row-11 with its corresponding full-BT baseline in Row-8, we see that this helps for gu→en, giving a further boost in performance of +0.8 to get a final BLEU score of **20.8**. To the best of our knowledge this outperforms the bilingual SoTA performance for gu→en (Bei et al., 2019). To summarize, except for hi→en, Iterative-BT helps improve Hinted BT significantly. For hi→en, even plain Iterative-BT does not help as seen in Row-3. Further investigating the cause of this result is delegated to future work.

6 Experiment Analysis

In this section, we analyse a few key aspects of the experiments described in the previous section.

6.1 Uniqueness of ta→en

We observed in Section 5.5 that Quality Tagging does not surpass the performance of the topk filtering strategy only for ta→en. In this section we investigate this observation further. Ta→en has two significant differences compared to the other two language pairs. First, from Table 5 we see that the bitext quality of ta→en is much poorer. Second, only 22% of the 23M BT data is present in topk-BT for ta→en, compared to 87% and 43% for hi→en and gu→en respectively. We posit that the large fraction of poor quality BT data interferes with the model learning from the bitext and high quality filtered BT data used in the topk-BT setting. In order to study this hypothesis, we train a model on a combination of 3 datasets: 630K of bitext, the 5M topk-BT, and 10M pairs randomly selected from the remaining 18M BT data. In total, we have 15M BT and 630K bitext pairs. To be consistent, we perform quality binning as in Section 5.5. In this setting, the model gets a BLEU score of **16.6**, outperforming the topk-BT method by $+0.2$ BLEU points. We repeat the above experiment by sampling 12 M noisy BT data (instead of 10M in the above set up). This drops the BLEU by 0.3 points.

Hence we see that the overarching trends of being able to learn from poor quality data via quality tagging also holds for ta→en. However ratio between good and poor quality BT data is important to achieve this improvement; especially when the bitext data is of poor quality. Understanding this interaction in more depth is left to future work.

6.2 Randomized Bin Assignment

In order to study the efficacy of Quality Tagging, we perform an experiment where instead of using *Equal Volume Binning* to choose bins (Row-6 of Table 4), we randomly assign every BT pair to one of four bins. We observe that BLEU of hi→en, gu→en and ta→en drops to 30.6, 16.8 and 15.9 respectively. In summary, we see that random bin assignment degrades performance of Quality Binning to almost match that of Tagged-BT.

6.3 Prediction of Translit Tags

As mentioned in Section 4, one of the key problems in cross-script NMT is to know when to translate, or transliterate a source word. In this section, we study the performance of our techniques in solving this problem. We pose the decision of translate vs transliterate as a binary classification problem as follows: comparing the source and target, we assign a binary label to every word in the source - *true* if it needs to be translated, *false* if it needs to be transliterated. Every NMT model we train is seen as a classifier that decides whether to translate/transliterate a word; we measure its F1 score that we call as 'word-level F1' (reported in Table 8). In Table 8, we see that models based on Translit Tags (Row-2) and Quality Binning + Translit Tags (Row-3) have equal/better F1 scores than the full-BT model (Row-1) across all languages. We also observe that adding quality tags, though unrelated to transliteration, helps improve the word-level F1.

#	Data	hi→en	gu→en	ta→en
1	bitext + full-BT	77.3	62.2	56.9
2	bitext + full-BT + translit-tags	77.8	62.2	58.9
3	bitext + full-BT + LaBSE quality tags + translit-tags	78.0	66.0	59.8
4	Correlation with BLEU	0.81	0.99	0.97

Table 8: %Word-level F1 scores of models in transliterating correct source words accurately. Row-4 shows Pearson's correlation of F1 scores with corresponding BLEU scores in Table 4.

We compute Pearson correlation between the word-level F1 scores with corresponding model BLEU scores (Row-4). As seen, there is a very strong correlation between these two variables, confirming that adding quality tags leads to better translate vs transliterate decisions. The combination of these two factors partially explain why the two hints lead to additive BLEU gains seen in Row-8 of Table 4.

6.4 Meta-Evaluation of Results

In this section, we perform meta-evaluation of our results using human evaluation and statistical significance tests as suggested by the guidelines in Marie et al. (2021). We compare each language pair's best non-iterative model (test system) and topk-BT model (base system) in Table 9. We first report their BLEU scores computed using SacreBLEU.

Then, we compute human evaluation scores for both the base and test systems (using the same metric described in Section 5.3). We have three human raters compare the base and test system translations using 500 randomly chosen source sentences from the test set. We report the difference in scores between the two systems (the Side-by-Side, i.e., SxS score) as the human evaluation metric. A SxS score of ± 0.1 between the two systems is considered significant. We see in Table 9 that hi→en and gu→en have sufficient SxS scores, whereas ta→en falls a little short of 0.1.

Finally, we perform statistical significance tests to compare the base and test systems (as described in Koehn (2004)). We create 1000 test sets with 500 random test datapoints each and calculate the two models' SacreBLEU scores. We use the resultant SacreBLEU scores to conduct T-tests[3]. For all three language pairs, we see significant T-statistics (reported in Table 9) which have p-values < 0.001.

[3]scipy.stats.ttest_ind

Metric	hi→en	gu→en	ta→en
Best (non-iterative) model's BLEU	31.6	17.9	16.6
topk-BT BLEU	31.2	16.0	16.4
Side-by-Side Human Eval.	0.11	0.19	0.05
T-statistic	11.05	64.03	8.43

Table 9: Meta-Evaluation of results. In this table, we compare each language pair's best non-iterative model (test system) and topk-BT model (base system) using three metrics - BLEU scores, SxS human evaluation scores, and T-statistics from statistical significance tests.

Data Used ↓	Bitext data size →			
	500k	200k	100k	50k
bitext	28.6	24.5	18.1	0.5
bitext + full-BT	33.7	31.2	27.4	1.5
bitext + full-BT +	36.6	34.3	30.9	3.1
LaBSE qual. tags	(+8.6%)	(+9.9%)	(+12.8%)	(+106.6%)

Table 10: Quality Tagging on simulated low resource scenarios of de→en

7 Issues in Low-Resource Settings

In this section, we discuss three issues that arise in low-resource settings, that are relevant to HintedBT. A language can be low resource if it (a) does not have enough bitext data (Section 7.1) or (b) is not well represented in open multilingual word / sentence embedding models (Section 7.2). Further, in scarce bitext settings, does having a large monolingual target corpus help HintedBT? (Section 7.3).

7.1 Low Bitext Quantity Simulation

Inspired by the experimentation methodology in Sennrich and Zhang (2019), we simulate different levels of low resource conditions using a high resource language pair German(de)→English(en). From the 38M bitext data points in de→en WMT 2019 news translation task, we randomly choose 500K, 200K, 100K, 50K bitext data points to simulate different low-resource scenarios. From 23M sentences of WMT 2013 Newscrawl's English monolingual data, we generate BT data and benchmark both full-BT and quality tagging on it. BT data is generated with en→de models trained on the restricted bitext for each setting. We use all of the 23M BT pairs since English monolingual data is easily available and we wanted to keep the setup as realistic as possible. Results in Table 10 clearly show that quality binning outperforms full BT under all scenarios. More interestingly, the

effectiveness of quality tagging increases as the low-resourcedness increases. This shows quality tagging is able to use all the data as full-BT, but more effectively, a very desirable characteristic in a low-resource setting.

7.2 Quality Metric for Extremely Low Resource Languages

LaBSE scoring (Feng et al., 2020) depends upon the availability of the pre-trained embedding model. Some very low-resource languages may not have multilingual embeddings or, even if present, may not have high quality embeddings. One alternative is to use round-trip-translation (Khatri and Bhattacharyya, 2020) and a syntactic comparison between the original target and the round-trip target. We use the Jaccard similarity index (Huang et al., 2008) between character tri-gram sets as the syntactic similarity measure. We call this measure the Bag of Trigram Jaccard or BoT-Jaccard in short.

We study BoT-Jaccard vs LaBSE in more detail in Appendix H. We summarize the results as follows. BoT-Jaccard has weaker correlation to human judgement of similarity compared to LaBSE. In our study of the failure patterns, most failures stem from the syntactic nature of the metric. Despite the above drawbacks of BoT-Jaccard over LaBSE, we see that it performs almost on par with LaBSE and hence is a very good alternative when LaBSE is not available. We repeat all our experiments with BoT-Jaccard and we see following improvements on the full-BT baseline. We get BLEU increases of 0.4 for hi→en, 2.8 for gu→en using quality tagging, and 1.4 for ta→en using topk-BT.

7.3 Does a larger monolingual corpus help?

In this section, we analyze if providing more BT data helps the model. We re-run HintedBT experiments from Section 5 with monolingual data from both Newscrawl 2013 and 2014, resulting in a total of 46M BT pairs. We report results in Table 11. For hi→en, quality tagging improves BLEU to **32.0** (an increase of 0.4 from our previous best of 31.6). For gu→en and ta→en, quality + translit tagging delivers performances of 18.2 and 16.1, +0.3 and +0.1 respectively from previous best experiments. This experiment shows while HintedBT does benefit from more data, the increase in performance does not commensurate to the large increase in volume of data.

#	Data	hi→en	gu→en	ta→en
	Using 23M BT	-	-	-
1	bitext + full-BT	30.9	15.0	14.1
2	Row-1 + LaBSE qual.tags	31.2	17.6	15.5
3	Row-2 + Translit-tags	31.6	17.9	16.0
	Using 46M BT	-	-	-
4	bitext + full-BT	31.3	14.9	14.4
5	Row-4 + LaBSE qual.tags	**32.0**	17.9	16.0
6	Row-5 + Translit-tags	31.3	**18.2**	**16.1**

Table 11: Experiments with a larger monolingual corpus. Rows 4-6 are directly comparable to rows 1-3.

8 Conclusion

In this work, we propose HintedBT, a family of techniques that adds hints to back-translation data to improve their effectiveness. We first propose *Quality Tagging* wherein we add tags to the source which indicate the quality of the source-target pair. We then propose *Translit Tagging* which uses tags on the target side corresponding to the translation/transliteration operations that are required on the source. We present strong experimental results over competitive baselines and demonstrate that models trained with our tagged data are competent with state-of-the-art systems for all three language pairs. The application of our techniques to multilingual models and to other generation techniques for back-translation (such as noised beam (Edunov et al., 2018)) are interesting avenues for future work.

Acknowledgements

We thank Gustavo Hernandez Abrego, Jason Riesa, Julia Kreutzer, Macduff Hughes, Preksha Nema, Sneha Mondal and Wolfgang Macherey for their early reviews and helpful feedback. We also thank the reviewers for their valuable and constructive suggestions.

References

Roee Aharoni, Melvin Johnson, and Orhan Firat. 2019. Massively multilingual neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3874–3884, Minneapolis, Minnesota. Association for Computational Linguistics.

Naveen Arivazhagan, Ankur Bapna, Orhan Firat, Dmitry Lepikhin, Melvin Johnson, Maxim Krikun, Mia Xu Chen, Yuan Cao, George Foster, Colin Cherry, et al. 2019. Massively multilingual neural

machine translation in the wild: Findings and challenges. *arXiv preprint arXiv:1907.05019*.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. 2018. Unsupervised statistical machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3632–3642, Brussels, Belgium. Association for Computational Linguistics.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Rachel Bawden, Nikolay Bogoychev, Ulrich Germann, Roman Grundkiewicz, Faheem Kirefu, Antonio Valerio Miceli Barone, and Alexandra Birch. 2019. The university of edinburgh's submissions to the wmt19 news translation task. *arXiv preprint arXiv:1907.05854*.

Chao Bei, Hao Zong, Conghu Yuan, Qingming Liu, and Baoyong Fan. 2019. Gtcom neural machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 116–121.

Nicola Bertoldi and Marcello Federico. 2009. Domain adaptation for statistical machine translation with monolingual resources. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 182–189, Athens, Greece. Association for Computational Linguistics.

Ondřej Bojar and Aleš Tamchyna. 2011. Improving translation model by monolingual data. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 330–336, Edinburgh, Scotland. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2018. Using monolingual data in neural machine translation: a systematic study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Brussels, Belgium. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2019. Using monolingual data in neural machine translation: a systematic study. *arXiv preprint arXiv:1903.11437*.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019a. Tagged back-translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019b. Tagged back-translation. *arXiv preprint arXiv:1906.06442*.

Peng-Jen Chen, Ann Lee, Changhan Wang, Naman Goyal, Angela Fan, Mary Williamson, and Jiatao Gu. 2020. Facebook ai's wmt20 news translation task submission. *arXiv preprint arXiv:2011.08298*.

Ryan Cotterell and Julia Kreutzer. 2018. Explaining and generalizing back-translation through wake-sleep. *arXiv preprint arXiv:1806.04402*.

Raj Dabre, Kehai Chen, Benjamin Marie, Rui Wang, Atsushi Fujita, Masao Utiyama, and Eiichiro Sumita. 2019. Nict's supervised neural machine translation systems for the wmt19 news translation task. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 168–174.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. *arXiv preprint arXiv:1808.09381*.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2020. Language-agnostic bert sentence embedding. *arXiv preprint arXiv:2007.01852*.

Vikrant Goyal and Dipti Misra Sharma. 2019. The iiit-h gujarati-english machine translation system for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 191–195.

Jiatao Gu, Yong Wang, Yun Chen, Victor O. K. Li, and Kyunghyun Cho. 2018. Meta-learning for low-resource neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3622–3631, Brussels, Belgium. Association for Computational Linguistics.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 151–157. AAAI Press.

Lars Hellsten, Brian Roark, Prasoon Goyal, Cyril Allauzen, Françoise Beaufays, Tom Ouyang, Michael Riley, and David Rybach. 2017. Transliterated mobile keyboard input via weighted finite-state transducers. In *Proceedings of the 13th International Conference on Finite State Methods and Natural Language Processing (FSMNLP 2017)*, pages 10–19.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine*

Translation and Generation, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Anna Huang et al. 2008. Similarity measures for text document clustering. In *Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand*, volume 4, pages 9–56.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2017. Improving low-resource neural machine translation with filtered pseudo-parallel corpus. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, pages 70–78, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Melvin Johnson, Mike Schuster, Quoc V Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, et al. 2017. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association of Computational Linguistics*, 5(1):339–351.

Marcin Junczys-Dowmunt. 2018. Dual conditional cross-entropy filtering of noisy parallel corpora. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 888–895, Belgium, Brussels. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA.

Jyotsana Khatri and Pushpak Bhattacharyya. 2020. Filtering back-translated data in unsupervised neural machine translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 4334–4339, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain control for neural machine translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 244–252, Brussels, Belgium. Association for Computational Linguistics.

Tom Kocmi and Ondřej Bojar. 2019. Cuni submission for low-resource languages in wmt news 2019. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 234–240.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

James Kuczmarski and Melvin Johnson. 2018. Gender-aware natural language translation.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The iit bombay english-hindi parallel corpus. *arXiv preprint arXiv:1710.02855*.

Guillaume Lample, Myle Ott, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2018. Phrase-based & neural unsupervised machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 5039–5049, Brussels, Belgium. Association for Computational Linguistics.

Bei Li, Yinqiao Li, Chen Xu, Ye Lin, Jiqiang Liu, Hui Liu, Ziyang Wang, Yuhao Zhang, Nuo Xu, Zeyang Wang, et al. 2019. The niutrans machine translation systems for wmt19. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 257–266.

Benjamin Marie, Atsushi Fujita, and Raphael Rubino. 2021. Scientific credibility of machine translation research: A meta-evaluation of 769 papers. *arXiv preprint arXiv:2106.15195*.

Austin Matthews, Waleed Ammar, Archna Bhatia, Weston Feely, Greg Hanneman, Eva Schlinger, Swabha Swayamdipta, Yulia Tsvetkov, Alon Lavie, and Chris Dyer. 2014. The cmu machine translation systems at wmt 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 142–149.

Toan Nguyen and David Chiang. 2018. Improving lexical choice in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 334–343, New Orleans, Louisiana. Association for Computational Linguistics.

Toan Q. Nguyen and David Chiang. 2017. Transfer learning across low-resource, related languages for neural machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 296–301, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Xing Niu, Michael Denkowski, and Marine Carpuat. 2018. Bi-directional neural machine translation with synthetic parallel data. In *Proceedings of the*

2nd Workshop on Neural Machine Translation and Generation, pages 84–91, Melbourne, Australia. Association for Computational Linguistics.

Venkatesh Balavadhani Parthasarathy, Akshai Ramesh, Rejwanul Haque, and Andy Way. 2020. The adapt system description for the wmt20 news translation task.

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, Belgium, Brussels. Association for Computational Linguistics.

Sree Harsha Ramesh and Krishna Prasad Sankaranarayanan. 2018. Neural machine translation for low resource languages using bilingual lexicon induced from comparable corpora. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 112–119, New Orleans, Louisiana, USA. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Parker Riley, Isaac Caswell, Markus Freitag, and David Grangier. 2020. Translationese as a language in "multilingual" NMT. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7737–7746, Online. Association for Computational Linguistics.

Mike Schuster and Kaisuke Nakajima. 2012. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152. IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40, San Diego, California. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich and Biao Zhang. 2019. Revisiting low-resource neural machine translation: A case study. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 211–221, Florence, Italy. Association for Computational Linguistics.

Jonathan Shen, Patrick Nguyen, Yonghui Wu, Zhifeng Chen, Mia X Chen, Ye Jia, Anjuli Kannan, Tara Sainath, Yuan Cao, Chung-Cheng Chiu, et al. 2019. Lingvo: a modular and scalable framework for sequence-to-sequence modeling. *arXiv preprint arXiv:1902.08295*.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. *arXiv preprint arXiv:1905.02450*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Lrec*, volume 2012, pages 2214–2218.

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*, pages 5998–6008.

Yiren Wang, ChengXiang Zhai, and Hany Hassan Awadalla. 2020. Multi-task learning for multilingual neural machine translation. *arXiv preprint arXiv:2010.02523*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Hayahide Yamagishi, Shin Kanouchi, Takayuki Sato, and Mamoru Komachi. 2016. Controlling the voice of a sentence in Japanese-to-English neural machine translation. In *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, pages 203–210, Osaka, Japan. The COLING 2016 Organizing Committee.

Biao Zhang, Philip Williams, Ivan Titov, and Rico Sennrich. 2020a. Improving massively multilingual neural machine translation and zero-shot translation. *arXiv preprint arXiv:2004.11867*.

Yuhao Zhang, Ziyang Wang, Runzhe Cao, Binghao Wei, Weiqiao Shan, Shuhan Zhou, Abudurexiti Reheman, Tao Zhou, Xin Zeng, Laohu Wang, et al. 2020b. The niutrans machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 338–345.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1568–1575, Austin, Texas. Association for Computational Linguistics.

A Topk-BT Baseline

We report our extensive experiments for finding the best topk-BT data here for both LaBSE and BoT-Jaccard based scoring of back-translated data pairs.

Data used	WMT test set	Dev set
LaBSE top 6.5M	29.6	20.1
LaBSE top 8M	30.4	20.2
LaBSE top 10M	30.6	20.5
LaBSE top 15M	30.7	20.3
LaBSE top 18M	**31.2**	20.3
LaBSE top 20M	**31.2**	**20.6**
Full-BT	30.9	20
BoT-Jaccard top 6.5M	30.4	20.8
BoT-Jaccard top 8M	30.7	**21.1**
BoT-Jaccard top 10M	30.4	20.5
BoT-Jaccard top 15M	**31**	20.6
BoT-Jaccard top 18M	30.6	20.5
BoT-Jaccard top 20M	30.7	20.3
Full-BT	30.9	20

Table 12: Grid search for best topk-BT data for hi→en

Data used	WMT test set	Dev set
LaBSE top 650k	13	22.3
LaBSE top 1M	13	23.3
LaBSE top 3.5M	15.4	26.3
LaBSE top 6.5M	15.7	27
LaBSE top 8M	**16.3**	26.9
LaBSE top 10M	16	**27.2**
LaBSE top 15M	16.1	26.8
Full-BT	15	25.8
BoT-Jaccard top 650k	12.2	21.5
BoT-Jaccard top 1M	12.8	22.3
BoT-Jaccard top 3.5M	15	25.7
BoT-Jaccard top 6.5M	15.4	26.7
BoT-Jaccard top 8M	16	26.9
BoT-Jaccard top 10M	**16.3**	**27**
BoT-Jaccard top 15M	15.5	26.7
Full-BT	15	25.8

Table 13: Grid search for best topk-BT data for gu→en

B Experimental Setup

We experiment with the following hyper-parameters -

- Number of encoder-decoder layers - 4, 6
- Number of attention heads - 12, 16
- Embedding dimensions - 768, 1024
- Hidden dimension - 1536, 8192

We choose the final model configuration described in Section 5.2 based on the dev-BLEU scores of the respective bitext models. However, further reduction of model size (reducing the number of attention heads, hidden dimension etc.) caused the models to underfit. The hi→en models have 375M parameters and gu→en and ta→en models have 283M parameters. Training was done using Tensorflow-Lingvo (Shen et al., 2019).

Note: For gu→en and ta→en, we randomly pick 200 pairs from each train source (from Table 2) and append them to the WMT newsdev set for better diversity.

Data used	WMT test set	Dev set
LaBSE top 2.5M	15.5	19
LaBSE top 5M	**16.4**	**19.9**
LaBSE top 8M	16	19.8
LaBSE top 10M	15.5	18.6
LaBSE top 15M	15.5	19.3
LaBSE top 20M	14.6	18.7
Full-BT	14.1	18.5
BoT-Jaccard top 2.5M	15.1	18.8
BoT-Jaccard top 5M	**16.5**	**19.6**
BoT-Jaccard top 8M	15.4	19.3
BoT-Jaccard top 10M	15.1	19.3
BoT-Jaccard top 15M	15.1	18.6
Full-BT	14.1	18.5

Table 14: Grid search for best topk-BT data for ta→en

C Human Evaluation of Data Quality

We ask human raters to evaluate the quality of source-target pairs (similar to Wu et al. (2016)). Quality scores range from 0 to 6, with a score of 0 meaning "completely nonsense translation",and a score of 6 meaning "perfect translation: the meaning of the translation is completely consistent with the source, and the grammar is correct". A translation is given a score of 4 if "the sentence retains most of the meaning of the source sentence, but may have some grammar mistakes", and a translation is given a score of 2 if "the sentence preserves some of the meaning of the source sentence but misses significant parts". These scores are generated by human raters who are fluent in both source and target languages.

The final human evaluation score of a set of n examples is given by the average of the n individual scores. When comparing two systems side-by-side, the difference between their two final scores quantifies the change in quality. In this case, a difference

of ± 0.1 is considered significant.

D FST Transliteration Models

To generate source to target language transliterations for *Translit Tagging*, we use FST transliteration models from Hellsten et al. (2017). Weighted Finite State Transducer (WFST) models are trained on individual word transliterations of native words from a set vocabulary, collected from 5 speakers amongst a large pool of speakers. These models are evaluated on annotated test sets for Hindi and Tamil, and they achieve 84% and 78% word-level accuracies respectively.

E Number of Bins : Quality Binning

We experiment with different number of bins in Equal Volume Binning for *Quality Tagging*. We show our experiments and corresponding dev-BLEU scores in Table 15.

Data	hi→en	gu→en	ta→en
bitext + full-BT	20.0	25.8	18.5
+ 3 LaBSE qual. tags	20.6	28.0	18.4
+ 4 LaBSE qual. tags	**20.8**	**28.4**	**18.6**
+ 5 LaBSE qual. tags	20.5	28.0	18.2

Table 15: Quality-Tagging Experiments with different number of bins

F Translit-tagging on the source side

In previous sections, we trained models with translit tags on the target side, hence enabling the models to predict whether or not transliteration should be done on the source. An alternative method is to provide these translit tags as *information* to the model, on the source side.

As we explain in Section 4, we require the target sentence to determine the translit tags. This is fine in the target-tagging case, since we do have access to the target while training; during inference, the model predicts the tag by itself. However, when we train a model with these tags on the *source*, it becomes necessary to provide this tag during inference as well - this renders this method is infeasible at test time. We conduct an oracle experiment where we assume the right tags are available from the target at test time. We report the results in Table 16. We see that for hi→en and ta→en, source tagging improves upon the full-BT baseline by +0.3; however for gu→en source tagging is worse by -0.2. For hi→en, source-tagging is better than target-tagging by +0.2; however for gu→en and ta→en, target-tagging is significantly better.

Data	hi→en	gu→en	ta→en
bitext + full-BT	30.9	15.0	14.1
(+ translit tags)	-	-	-
Source-Tagged	31.2	14.8	14.4
Target-Tagged	31.0	15.2	15.0

Table 16: Source side Translit-tagging on full-BT data

G Alternate Experiments for hi→en

In our hi→en experiments, we use the IIT Bombay en-hi Corpus (Kunchukuttan et al., 2017) with 1.68M source-target pairs as the training dataset. In this section, we repeat our HintedBT experiments with the original training set from WMT-2014, which has 271k source-target pairs. We report test scores on the WMT-2014 hi→en newstest set in Table 17.

Modeling Methodology	Test
bitext	10.3
bitext + full-BT	25.5
bitext + full-BT + *LaBSE* quality tags	**27.4**
bitext + full-BT + *LaBSE* quality tags + translit-tags	27.0

Table 17: Experiments with hi→en WMT-2014 train set.

H Comparison of BoT-Jaccard against LaBSE as a Quality Metric

We run all the experiments in Section 5 with BoT-Jaccard scores in the place of LaBSE scores. We present results in Table 18. We see for hi→en, the topk-BT baseline is lower than the full-BT baseline, whereas for gu/ta→en, topk-BT is higher. For hi/gu→en, the BoT-Jaccard score based quality tagging gives competent results, whereas for ta→en, the topk-BT model remains the best result.

To better understand patterns of LaBSE or BoT-Jaccard mistakes in evaluating quality for parallel data, we manually annotate back-translations for hi→en where the metrics oppose each other. We select 200 random instances where,

$$abs(\text{BoT-Jaccard} - \text{LaBSE}) > 0.2$$
$$\text{and } min(\text{BoT-Jaccard}, \text{LaBSE}) < 0.5$$

We manually annotate which metric is correct, and the reason for the other metric's failure. We present the analysis in two parts, one where BoT-Jaccard score is higher than LaBSE, and the other where

Data	hi→en	gu→en	ta→en
bitext	19.5	8.4	11.3
bitext + full-BT	30.9	15.0	14.1
bitext + *Jacc.* topk-BT	30.7	16.3	**16.5**
bitext + full-BT + *Jacc.* quality tags	**31.3**	**17.8**	15.7
bitext + *Jacc.* topk-BT + translit-tags	30.7	16.1	15.8
bitext + full-BT + *Jacc.* quality tags + translit-tags	31.0	17.7	16.3

Table 18: Performance of models on WMT test sets, using BoT-Jaccard scoring. These results are directly comparable to corresponding rows in Table 4.

Quality Metric	hi→en	gu→en	ta→en
BoT-Jaccard	0.127	0.306	0.245
LaBSE	0.262	0.399	0.314

Table 19: Spearman's correlation coefficient for quality metrics against human judgements of quality (1500 samples for each). p-value < 0.001 for all scores.

LaBSE is higher than BoT-Jaccard. In Table 20 and Table 21 we present the categorizations of mistakes made by either method. Figure 3 shows examples of source sentences, their back translations, and round trip translations which are referred to in the analysis.

Ex. #	Source	Back Translation	Round Trip Translation
1	Data Library	ऑकडा लाइब्रेरी	Data Library
2	Derby City	डर्बी सिटी City name (optional, probably does not need a translation)	Derby City
3	This is how it will work.	तत्काल वह नीचे कूद गया। *(Literal translation: He immediately jumped down)*	This is how it works.
4	Skilled Labor	कुशल प्रसव *(प्रसव is used for pregnancy related labor, so incorrect in the context)*	Skilled Labor
5	Nightingale	बुलबुल	Nightingale
6	Mild special needs	हलकी विशेष आवश्यकताएं	Light Requirements
7	My dignity.	मेरी इज्जत।	Thank you so much.
8	Extra aid	अतिरिक्त सहायता	Additional Support
9	10 - Milk shake	10 -दूध हिलना *(Literal translation: 10 -milk shaking)*	It is like moving
10	Religion of peace.	शांति धर्म।	Shanti Dharma.

Figure 3: Examples from the qualitative analysis of Jaccard and LaBSE scores. Text in italics are added as comments. Everything else is part of system output.

Reason	**#**	**Explanation**
LaBSE cannot capture similarity for transliterations	45	This is probably because LaBSE has not been trained on parallel data which contains transliterations. Entities in particular are often transliterated (transcribed in Devanagari), but sometimes even common words like “library” are used through transliteration rather than translation, in Hindi. Row 1 in Figure 3 is an example for this.
Mistake in BT fixed by RTT deceives Jaccard	40	There are further three categories of mistakes in BT here. 1. The first is some random noise added to the BT probably stemming from the training data. The phrase "City name (optional, probably does not need a translation)" on Row 2 in Figure 3 is generated by the BT model and corrected by the RTT model. 2. Second, is a completely irrelevant BT that is somehow corrected by the RTT like Row 3 in Figure 3. This might also be due to faulty training data for the BT and RTT models. 3. The last mode of failure is where the BT is wrong because it uses a wrong synonym for translating a source word like Row 4 in Figure 3 where the word for pregnancy labor is used for translating the phrase “skilled labor".
LaBSE misses semantic similarity in source and BT	15	This is the least common mode of failure and might point to some gaps in LaBSE training for this language pair (not trained with enough data to cover rare synonyms or formulations). On Row 5 in Figure 3, LaBSE does not recognize the correct translation for “Nightingale".

Table 20: Categorization and number of examples where LaBSE >> Jaccard.

Reason	#	Explanation
Model translating BT to RTT makes a mistake and deceives Jaccard	46	In the most common case, the Back Translation is correct, and this is correctly captured by LaBSE. However, the model translating BT to RTT makes a mistake, and therefore fools Jaccard on this instance. Row 6 in Figure 3 is an example of slight difference in meaning between the correct BT and the RTT. In Row 7 the BT is correct, however the RTT is completely random.
Synonyms used in RTT which preserves meaning but deceives Jaccard	41	In the second most common case, both the BT and RTT seem to have the same meaning as the original source sentence. However, the model translating BT to RTT uses synonyms of words in the source and therefore results in a low Jaccard score. Row 8 in Figure 3 is an example of this.
Mistake in both BT and RTT - wrongly marked as close by LaBSE	9	In this case, there is a slight mistake in meaning when source is translated to BT and it is further compounded by RTT. However, LaBSE marks the source and BT as close, which is incorrect. Row 9 in Figure 3 is an example of this.
Reverse model transliterates, which deceives Jaccard	5	Finally, in some examples, the reverse model transliterates the BT instead of translating it, resulting in low Jaccard scores. Row 10 in Figure 3 is an example of this.

Table 21: Categorization and number of examples where Jaccard $>>$ LaBSE.