Large-scale Machine Translation for Indian Languages in E-commerce under Low Resource Constraints

Amey Patil, Nikesh Garera
Flipkart
{amey.patil,nikesh.garera}@flipkart.com

Abstract

The democratization of e-commerce platforms has moved an increasingly diversified Indian user base to shop online. We have deployed reliable and precise large-scale Machine Translation systems for several Indian regional languages in this work. Building such systems is a challenge because of the low-resource nature of the Indian languages. We develop a structured model development pipeline as a closed feedback loop with external manual feedback through an Active Learning component. We show strong synthetic parallel data generation capability and consistent improvements to the model over iterations. Starting with 1.2M parallel pairs for English-Hindi we have compiled a corpus with **400M+** synthetic high quality parallel pairs across different domains. Further, we need colloquial translations to preserve the intent and friendliness of English content in regional languages, and make it easier to understand for our users. We perform robust and effective domain adaptation steps to achieve colloquial such translations. Over iterations, we show **9.02** BLEU points improvement for English to Hindi translation model. Along with Hindi, we show that the overall approach and best practices extends well to other Indian languages, resulting in deployment of our models across 7 Indian Languages.

1 Introduction

As one of the largest e-commerce platform, we support a very diverse user base in terms of regional languages. Product Descriptions, Catalog Attributes, and Product Reviews help customers understand and compare various products available on the platform. For the growing user-base in India with non-English background, providing this information in regional Indian languages makes their shopping experience more informative and friendly. With only 10% of the Indian population being versed in English [1], vernacular support is vital for the platform and its diverse users. In this work, we develop Machine Translation System to translate the available product data from English to regional languages to address this problem. Given the size of the Product Catalog and user base, the volume of the data to be translated is in the order of 100s of millions. This poses a challenge to build Translation Systems that are robust, reliable, and precise at scale.

The low resource nature of Indian Languages[2] is another challenge for data-hungry deep networks such as Transformer(Vaswani et al., 2017). Given a large enough parallel corpus, the Transformer model can learn the inter-lingual mappings very well, even for very long sequences. These models can generate human-level precision translations for some resource-rich European languages(Popel et al., 2020). So theoretically, if we can get a large enough parallel corpus for Indian languages, we can solve the Automatic Machine Translation for Indian languages.

We build a training pipeline that can take monolingual corpus(abundantly available from public and in-house sources) and generate a high-quality synthetic parallel corpus. This is an efficient and effective approach, especially when paired with the Active Learning component over model iterations. For Hindi, starting with 1.2M parallel examples, we have compiled over 400M synthetic parallel examples with numerous model iterations.

Translation is an inherently one-to-many task where a single text can have various correct translations. The domain gap between the e-commerce domain and public domain (news, government sites, Wikipedia, books, etc.) is significant. To showcase this, Figure 1 has colloquial and non-colloquial

[1] https://en.wikipedia.org/wiki/2011_Census_of_India#Language_demographics

[2] The most extensive parallel corpus has 8.56M English-Hindi translation pairs from Samanantar Dataset(Ramesh et al., 2021)

Proceedings of EMNLP 2022 Industry Track, pages 637–644
December 9–11, 2020.

English input	Colloquial Translation	Non-Colloquial Translation
A good quality product for you	आपके लिए एक अच्छी क्वालिटी का प्रोडक्ट	आपके लिए एक अच्छी गुणवत्ता का उत्पाद
It is perfect for active wear, road ripping or for casual day out.	यह एक्टिव वियर, रोड रिपिंग या कैजुअल डे आउट के लिए परफेक्ट है।	यह सक्रिय पहनने, सड़क पर दौड़ने या आकस्मिक दिन के लिए एकदम सही है।
Decent product at this price segment	इस प्राइस सेगमेंट में अच्छा प्रोडक्ट है।	इस कीमत वर्ग में बेहतरीन उत्पाद।
And to Filter Ultraviolet Rays.	और अल्ट्रावायलेट रे को फ़िल्टर करने के लिए है।	और पराबैंगनी किरणों को छानने के लिए।
Q: Sound quality is how?	प्रश्न: साउंड क्वालिटी कैसी है?	प्रश्न: ध्वनि की गुणवत्ता है कैसे?
clearity is just awesome in it.	क्लैरिटी बस कमाल है इसमें।	इसमें स्पष्टता बस कमाल है।

Figure 1: Both colloquial and non-colloquial translations are correct, but for E-commerce platform we need more colloquial translation styles.

Hindi translations for a source sentence in English. Both of these translations are correct, but as an e-commerce platform, we refrain from using non-colloquial and infrequently used words as it decreases the appeal of the information from the colloquial e-commerce English domain.

Based on the final training steps, translation models can generate appropriate translations at inference. We fine-tune the model only using the colloquial in-domain data with robust domain adaptation steps to get more colloquial translations.

Our contributions in this paper are as follows:

- **Synthetic Parallel Corpus Generation**: With the help of sub-modules, we generate a vast amount of high-quality parallel corpus solving for low-resource Indian Languages.
- **Iterative Model Training Pipeline**: With the help of data cleaning and filtering modules, we showcase how we iteratively improve the Translation models significantly with Active Learning steps.
- **Large-Scale High Precision and Colloquial Models**: Finally, we provide large-scale Machine Translation models with high precision and domain-adapted colloquial styles for several Indian Languages.

2 Related Work

Transformers (Vaswani et al., 2017) are widely used architecture for seq2seq tasks. Along with Unigram-based subword tokens, the fully attention-based model performs very well for Translation tasks, even for longer sequences. Translation is a well-explored area, and even for low-resource settings, significant work has already been done. Along the lines of data gathering - collecting parallel corpora (Ramesh et al., 2021), mining multilingual sets and retrieving parallel entries (Tran et al., 2020), iterative cross-lingual alignments (Philip et al., 2021) has been explored. Zhang et al. (2020), showed parallel corpus filtering on web crawled data.

Transfer Learning is also a convenient approach to improve final model performance in low resource settings. (Rothe et al., 2020) explored leveraging large language models trained on unlabelled data for translation tasks. This approach works well only if we have strong pre-trained models. For Indian language settings, this is typically not the case. Also, synthetic data generation is very inefficient without active learning. (Imankulova et al., 2019) shows that translation models can help with pseudo labeling, but this improvement saturates without external feedback. (Peris and Casacuberta, 2018) has explored an active learning framework for machine translation. Gupta et al. (2021) investigate the active learning methods for Machine Translation in Indian Languages settings. Lample et al. (2017) even shows that completely unsupervised Machine Translation is possible using just monolingual data. But these practices don't work in large-scale settings. Given a large amount of good quality parallel data, supervised methods still beat other weak methods. Especially for production settings, there has not been much exploration done at large-scale systems starting with low resource settings.

3 Overall Pipeline

We use Transformer encoder-decoder model with 6 encoder layers and 6 decoder layers with hidden size of 512. We use 32,000 unigram subword tokens trained on data from all domains. This configuration has 93M parameters. As a pre-processing step, we split long paragraphs into sentences and translate independent sentences using the Transformer model.

3.1 Datasets Used

We use all publicly available parallel corpus from various domains within commercial licensing restrictions. Also, we conduct internal operations for parallel corpus creation for in-domain sampled datasets from the Product Descriptions, Catalog Attributes, Search, and Product Reviews. This operation is costly and is only done with the Active Learning step. Apart from parallel corpus, our pipeline heavily relies on synthetic data generation, for which we use publicly available monolingual corpus from general domain compiled from various sources (Wenzek et al., 2020; Abadji et al., 2022; Barrault et al., 2019). The details for the datasets are listed in table 1.

Language	Public Parallel Corpus	In-House Tagged Corpus	Public Mono. Corpus
Hindi	0.89M	0.28M	167M
Tamil	0.86M	0.46M	80M
Telugu	0.32M	0.44M	23M
Bengali	2.13M	0.44M	79M
Marathi	0.448M	0.16M	15M
Malayalam	0.61M	0.16M	32M
Kannada	0.05M	0.20M	18M

Table 1: Monolingual Datasets used in Synthetic Data Generation along with Parallel Corpus

3.2 Monolingual Data Processing

To generate synthetic parallel corpus, we use well-known Back-Translation methods (Sennrich et al., 2016) to translate Indic monolingual corpus back to English. The corpus we use from the public domain is already curated and cleaned. So cleaning Indic monolingual data is easy with some basic text-normalization, rare character filtering, punctuation fixes, etc.

Apart from back-translations, we also use Forward Translations, where we translate monolingual source text to the target language with an imperfect translation model. Forward Translations are a crucial part of our training pipeline. But the quality of synthetic pairs heavily impacts the final model training. Domains such as English Reviews and Search queries can be very noisy with spelling errors, punctuation errors, case errors, etc. While generating translations for these noisy entries, the quality of the translations is limited by the noisy input itself. Hence before using monolingual data for synthetic parallel data generation, we filter out unclean English texts from the corpus using the pipeline as shown in Fig. 2. We use BERT based classifier model to detect noisy texts from the monolingual corpus. To improve the Translation model robustness, we (1) Correct some of the noisy data filtered out from monolingual corpus to get translations even for noisy text inputs and (2) introduce noise to already clean input texts. We use In-House Transformer-based Encoder-Decoder Spell Correction models to correct unclean texts for search queries and reviews. And as spell correct models have low precision (benchmarks detailed in table 2) we again filter out unclean data from spell corrected set as shown in Fig. 2. We add pairs <*noisy text, translation from cleaned corrected text*> as the translation pairs in generated training data.

Data Stream	General Domain	Search	Reviews
F1 score - Noisy text classification	95.03	90.24	92.55
Spell Correct Rate	-	80.53%	55.75%

Table 2: Monolingual Data Cleaning. (Spell Correction rate is the percent of unclean text model corrects properly)

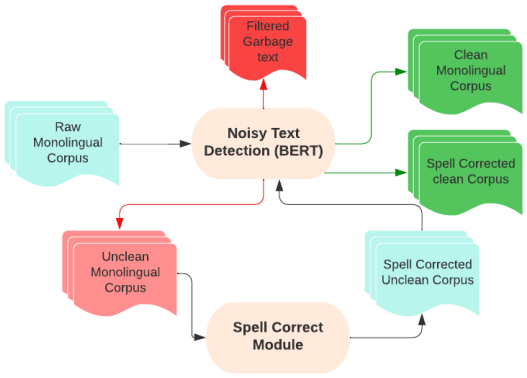

Figure 2: Monolingual Data Cleaning and Spell Correction pipeline.

3.3 Translation Quality Estimation

We monitor and filter imperfect parallel pairs with two methods:

- **Translation model Uncertainty score**: The transformer model uses predictions from a

softmax layer to generate each token output. The output of this layer is the probability distribution over the vocabulary for each token. When the probability of the predicted token is low - the model is more uncertain about the token prediction and vice versa. We aggregate this metric over the entire output sequence and normalize it for the output length to get a final uncertainty score for a translation.

- **Independent BERT based Quality estimation**: Given a source and target sequence, we train a multilingual BERT (Devlin et al., 2018) based classifier, which predicts if the sequences are perfect parallel pairs. The BERT-based classifier model is trained on a set of correct translation pairs (pooled from available high-quality manual translation pairs) and noise-induced pairs from the correct translation pairs with multiple levels of translation errors. To get the final translation score, we pass both the source-target and target-source combination of pairs to a pre-trained BERT encoder and use concatenated context for the classification head.

$$quality\_score(x_{1..T}, y_{1..T'}) =$$

$$\text{h}([\text{B}(\text{x}_{1..T}, y_{1..T'}), B(y_{1..T'}, x_{1..T})]) \quad (1)$$

Model	Precision (good trans.)	Recall (bad trans.)	Overall F1 score
Uncertainty Scoring	0.8889	0.8375	0.8554
BERT Translation Scoring	0.8091	0.6750	0.8166
Ensemble	**0.8899**	**0.8500**	0.8264

Table 3: Translation Quality Estimation Benchmark.

As the Translation model Uncertainty score can still be biased toward the erroneously predicted tokens, the independent translation scoring is a good supplement for data filtering. We use an ensemble of two translation quality estimation methods and reject translations setting up high rejection recall. The evaluation scores for both models and ensemble are detailed in Table 3. The final filtered data counts are detailed in table 4 for Hindi language. As expected, rejected synthetic translations are very high for search and reviews set as the stream has very noisy inputs.

Monolingual Dataset	Data Language	Dataset Size	Final Filtered Syn. Parallel Corpus Size
CC-100	Hindi	94.08M	83.09M
OSCAR	Hindi	12.85M	10.77M
news-crawl	Hindi	48.86M	45.11M
Wikipedia	Hindi	1.85M	1.61M
Our Product Descriptions	English	71.32M	70.45M
Our Catalog Attributes	English	64.82M	56.73M
Our Reviews	English	65.80M	44.79M
Our Search	Hindi	29.81M	29.14M
Our Search	English	218.71M	83.03M
Total	-	**608.1M**	**424.72M**

Table 4: Monolingual datasets used and back-translated or forward translated dataset size and filtered synthetic corpus size.

3.4 Pipeline with Active Learning

To generate the synthetic parallel corpus and train Translation models using this corpus, we use a pipeline demonstrated in figure 3 in an iterative manner. The detailed algorithm is mentioned in Algo. 1.

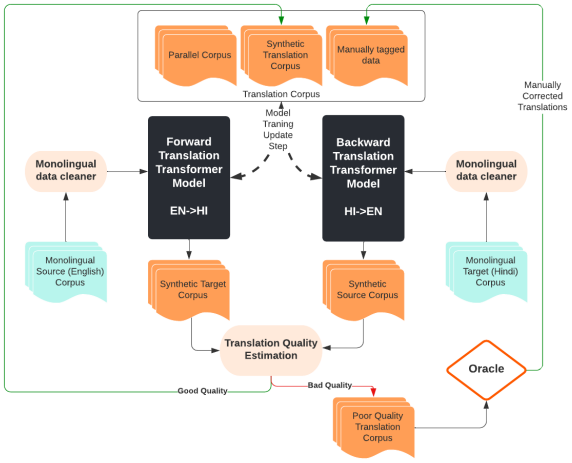

Figure 3: Model Training and Synthetic Data Generation pipeline

We start with English to Indic and Indic to English translation models trained with publicly available and in-house parallel corpus. This base corpus provides good first-version translation models for our iterative pipeline. Iteratively, we process more monolingual data through the pipeline and add good quality synthetic corpus to the training set. The monolingual in-domain text, which the model can not translate accurately, detected by the

Algorithm 1: Training + Data generation pipeline.

Models : M_f (Forward Translation model)
M_b (Backward Translation model)
M_n (Noisy text detection BERT)
M_s (Spell correct model)
M_q (Translation Quality Est.)
Data : P (Existing parallel corpus)
C_s (mono. source lang. corpus)
C_t (mono. target lang. corpus)

begin
  M_f = TransformerTraining(P)
  M_b = TransformerTraining(P)
  repeat
    $C_{s\_clean}, C_{s\_noisy} = M_n(C_s)$
    $C_{s\_corr} = M_{spell}(C_{s\_noisy})$
    $C_{s\_corr\_clean}, C_{s\_corr\_noisy} = M_n(C_{s\_corr})$
    C'_{s_clean} = Translate(C_{s_clean}, M_f)
    $C'_{s\_corr\_clean}$ = Translate($C_{s\_corr\_clean}, M_f$)
    C'_t = Translate(C_t, M_b)
    $S = (C_{s\_clean}, C'_{s\_clean}) + (C_{s\_corr\_clean}, C'_{s\_corr\_clean}) + (C'_t, C_t)$
    $S_{good}, S_{poor} = M_{quality}(S)$
    S_{s_poor} = sample(S_{poor})
    S_{corr} = oracle(S_{s_poor})
    $TR = S_{good} + S_{corr} + P$
    M_f = TransformerTraining(TR)
    M_b = TransformerTraining(TR)
    C_s = collect()
    C_t = collect()
  until *Satisfactory precision achieved*;
end

Translation Quality Estimation module, is pooled, and a diverse batch is sampled from this set to get corrected by manual annotators. This batched Active Learning is crucial in the iteration and makes the forward translations feasible. While re-training the model in the next model iteration, we have filtered good synthetic translations generated by the model and manual translations instead of imperfect translations the model produces. This is an overall translation corpus quality update; hence we train improved Translation models in each iteration.

3.5 Domain Adaptation

As we need colloquial translations in the output, we have to fine-tune the pre-trained models on all domain corpus using just the in-domain colloquial dataset. As evident from Table 5, BLEU scores jump sharply when the model is fine-tuned on the in-domain small training set. This shows that domain gap with general domain and e-commerce colloquial domain is significant. In-domain Forward Translations(forward translated in-domain monolingual corpus) are crucial in this step as the cleaned

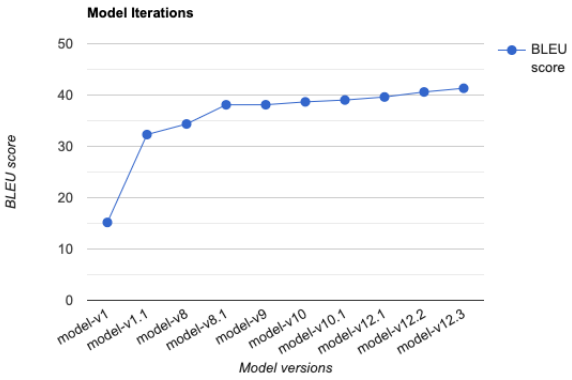

Figure 4: Snapshot of selected models. Iterations vs Product Description BLEU scores for English-Hindi Translation.

Model	Training Steps & Corpus	PD BLEU
Google	-	36.27
Azure	-	29.29
IndicTrans	Samanantar Dataset	31.15
model-v1	Public parallel corpus	15.18
model-v1.1	v1 =>In-domain fine-tuning	32.3
model-v2	Public Parallel Corpus + 50M back-translations	32.3
model-v2.1	v2 =>In-domain fine-tuning	37.59
model-v8	Public Parallel Corpus + 150M back-translations	34.36
model-v8.1	v8 =>In-domain fine-tuning	38.1
model-v9	v8 =>Forward translations	38.11
model-v10	v8 =>Filtered Forward translations	38.56
model-v12	Public Parallel Corpus + 150M back-translations + Filtered Forward translations	37.22
model-v12.1	v12 =>In-domain fine-tuning	39.62
model-v12.2	v12 =>Active Learning (+50k)	40.6
model-v12.3	v12 =>Active Learning (+80k)	41.32

Table 5: Hindi Product Description BLEU scores

and filtered high-quality forward translations help bridge the domain gap and provide much more capable pre-trained models. This ensures that the model does not go through over-fitting or catastrophic forgetting(for the in-domain set), and we get a more robust and reliable model at scale. Table 1 has some examples where our model produces more colloquial translations and refrains from using non-colloquial and non-friendly Hindi words.

3.6 Model Iterations

As evident from Table 4, the BLEU scores are drastically improved in each synthetic data addition step. The best model is improved by **+9.01** BLEU scores over the v1.1 model which does not use any synthetic corpus. The size of back-translated cor-

Language	EN->X PD		EN->X WAT21		X->EN WAT21		Our Translation Accuracy	
	Ours	Google	Ours	Google	Ours	Google	PD	Catalog Attributes
Hindi	**41.32**	36.27	**32.95**	32.5	**37.85**	36.7	95.76%	97.39%
Tamil	**44.83**	31.86	**10.65**	8.98	**25.37**	23.51	94.36%	95.54%
Telugu	**39.69**	30.78	**4.34**	4.21	**26.28**	25.66	90.87%	94.31%
Bengali	**30.65**	24.33	**7.56**	5.05	**22.51**	20.52	98.87%	91.13%
Marathi	**37.37**	28.86	**12.96**	12.6	**28.07**	26	82.05%	95.14%
Kannada	**31.32**	24.19	**12.85**	12.9	**29.61**	24.75	90.38%	96.94%
Malyalam	**30.57**	27.83	**5.09**	10.6	**28.32**	27.2	-	93.32%

Table 6: BLEU scores comparing best public API and Manual Translation Accuracy for our Product Descriptions(PD) and Catalog Attributes.

pus is also impactful even in the range of 10s of million entries, as more data helps significantly. Forward translations are very critical part of the synthetic corpus, as theoretically the quality of forward translations is limited by the performance of the translation model itself used to generate the forward translations. This is where translation quality estimation plays a crucial role for filtering out low quality translations. From Table 4, the model v9 performs very similar to v8.1, which is used to generated forward translations for a large set. But once we filter out imperfect translations, even forward translations show an improved final v10 model. Finally, the additional small set of manual translations generated from Active Learning step over these imperfect translations provides even better v12.* models.

Hindi Model	Test set	Good Trans.	Can be better Trans.	Bad Trans.
Catalog, PD Model	PD	53%	42%	**5%**
Google	PD	14%	51%	35%

Table 7: English to Hindi Translation evaluation for Product Descriptions(PD)

4 Results and Discussion

We benchmark our models on manually annotated Product Descriptions(PD) test set along with public Indic WAT21 benchmark(Nakazawa et al., 2021) in table 6. We consistently show better BLEU scores on all test sets than Public translation API(Google).

We define the Translation Accuracy i.e., the rate at which the translation is acceptable with only minor errors(percent excluding bad cases), is very high across all languages. This allows us to deploy the Translation Systems in large-scale, highly precise settings. Table 7 shows the exact figures for manual evaluation for English to Hindi catalog translations. Our models show remarkably low bad translation cases and very high, ($> 50\%$) gold standard translations. The huge domain gap between e-commerce and general domains leads to poor evaluation results for Google as it produces consistent non-colloquial words and is not adapted to the domain.

The training pipeline has consistently shown better translation models throughout the model iterations paired with Active Learning, adding more monolingual data and filtered high-quality synthetic parallel translations. As evident from the plot 4, the addition of more synthetic data in pretraining, the addition of forward translation for pretraining as well as domain adaptation, model retraining from scratch with higher quality corpus and better pipeline sub-modules, and active learning steps show very significant improvements at each stage. Starting from 32.3 BLEU score, we have reached 41.32 BLEU score, which is a massive improvement just using a few active learning steps and synthetic corpus updates.

4.1 Deployment and Business Impact

Currently the Translation models for all the languages are deployed in batch-prediction mode on CPU inference system. While translating the catalog data or updating the translation models, we trigger the deployment pipeline and update the offline batch-predictions in the database.

The primary metric used to determine the impact of this deployment is conversion and cost savings. We have seen +11 bps improvement in conversion and significant cost savings through 100% automated translations via our system across various

languages.

5 Conclusion

In this work we have shown that synthetic parallel corpus generation and data filtering is a viable option to train large-scale translation models in low-resource settings. Also we show that Active Learning can consistently improve the model. We build very robust, large-scale models which work very precisely on our In-domain data and also outperform Google on public general domain benchmarks consistently. We also show how building colloquial models are important for ease of understanding, and we also show that our overall approach and best practices extend well to multiple Indian languages.

Limitations

The proposed training pipeline heavily relies on synthetic translations. In some cases(for example, Assamese has only <1M monolingual text), there is not enough data, and the initial model itself can not be appropriately trained, which makes the entire pipeline ineffective. Data efficiency is a considerable challenge in low-resource settings.

The pipeline uses several Language Model based sub-modules for data-cleaning, translation quality estimation, etc., which also impact the pipeline capability, and it might get cumbersome to manage and update many modules.

References

Julien Abadji, Pedro Ortiz Suarez, Laurent Romary, and Benoît Sagot. 2022. Towards a Cleaner Document-Oriented Multilingual Crawled Corpus. *arXiv e-prints*, page arXiv:2201.06642.

Loïc Barrault, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 conference on machine translation (WMT19). In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 1–61, Florence, Italy. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Kamal Gupta, Dhanvanth Boppana, Rejwanul Haque, Asif Ekbal, and Pushpak Bhattacharyya. 2021. Investigating active learning in interactive neural machine translation. In *Proceedings of Machine Translation Summit XVIII: Research Track*, pages 10–22, Virtual. Association for Machine Translation in the Americas.

Aizhan Imankulova, Takayuki Sato, and Mamoru Komachi. 2019. Filtered pseudo-parallel corpus improves low-resource neural machine translation. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 19:1–16.

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only.

Toshiaki Nakazawa, Hideki Nakayama, Isao Goto, Hideya Mino, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Shohei Higashiyama, Hiroshi Manabe, Win Pa Pa, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, Yusuke Oda, Katsuhito Sudoh, Sadao Kurohashi, and Pushpak Bhattacharyya, editors. 2021. *Proceedings of the 8th Workshop on Asian Translation (WAT2021)*. Association for Computational Linguistics, Online.

Álvaro Peris and Francisco Casacuberta. 2018. Active learning for interactive neural machine translation of data streams. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 151–160, Brussels, Belgium. Association for Computational Linguistics.

Jerin Philip, Shashank Siripragada, Vinay P. Namboodiri, and C. V. Jawahar. 2021. Revisiting low resource status of indian languages in machine translation. In *8th ACM IKDD CODS and 26th COMAD*. ACM.

Martin Popel, Marketa Tomkova, Jakub Tomek, Łukasz Kaiser, Jakob Uszkoreit, Ondřej Bojar, and Zdeněk Žabokrtský. 2020. Transforming machine translation: a deep learning system reaches news translation quality comparable to human professionals. *Nature Communications*, 11(4381):1–15.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Mahalakshmi J, Divyanshu Kakwani, Navneet Kumar, Aswin Pradeep, Srihari Nagaraj, Kumar Deepak, Vivek Raghavan, Anoop Kunchukuttan, Pratyush Kumar, and Mitesh Shantadevi Khapra. 2021. Samanantar: The largest publicly available parallel corpora collection for 11 indic languages.

Sascha Rothe, Shashi Narayan, and Aliaksei Severyn. 2020. Leveraging pre-trained checkpoints for sequence generation tasks. *Transactions of the Association for Computational Linguistics*, 8:264–280.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Improving neural machine translation models with monolingual data.

Chau Tran, Yuqing Tang, Xian Li, and Jiatao Gu. 2020. Cross-lingual retrieval for iterative self-supervised training.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

Guillaume Wenzek, Marie-Anne Lachaux, Alexis Conneau, Vishrav Chaudhary, Francisco Guzmán, Armand Joulin, and Edouard Grave. 2020. CCNet: Extracting high quality monolingual datasets from web crawl data. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4003–4012, Marseille, France. European Language Resources Association.

Boliang Zhang, Ajay Nagesh, and Kevin Knight. 2020. Parallel corpus filtering via pre-trained language models.