Samsung R&D Institute Poland submission to WAT 2021 Indic Language Multilingual Task

Adam Dobrowolski, Marcin Szymański, Marcin Chochowski, Paweł Przybysz
Samsung R&D Institute, Warsaw, Poland
{a.dobrowols2, m.szymanski, m.chochowski, p.przybysz} @samsung.com

MultiIndicMT: An Indic Language Multilingual Task
Team ID: SRPOL

Abstract

This paper describes the submission to the WAT 2021 Indic Language Multilingual Task by Samsung R&D Institute Poland. The task covered translation between 10 Indic Languages (Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil and Telugu) and English.

We combined a variety of techniques: transliteration, filtering, backtranslation, domain adaptation, knowledge-distillation and finally ensembling of NMT models. We applied an effective approach to low-resource training that consist of pretraining on backtranslations and tuning on parallel corpora.

We experimented with two different domain-adaptation techniques which significantly improved translation quality when applied to monolingual corpora. We researched and applied a novel approach for finding the best hyperparameters for ensembling a number of translation models.

All techniques combined gave significant improvement - up to +8 BLEU over baseline results. The quality of the models has been confirmed by the human evaluation where SRPOL models scored best for all 5 manually evaluated languages.

1 Introduction

Samsung R&D Poland Team researched effective techniques that worked especially well for low-resource languages: transliteration, iterative back-translation followed by tuning on parallel corpora. We successfully applied these techniques during the WAT2021 competition (Nakazawa et al., 2021). Especially for the competition we also applied custom domain-adaptation techniques which substantially improved the final results.

Most of the applied techniques and ideas are commonly used for works on Indian languages machine translation (Chu and Wang, 2018) (Dabre et al., 2020).

This document is structured as follows. In section 2 we describe the sources and techniques of corpora preparation used for the training. In sections 3 and 4 we describe the model architecture and techniques used in training, tuning and ensembling and finally Section 5 presents the results we gained on every stage of the training.

All trainings were performed on Transformer models. We used standard Marian NMT [1] v.1.9 framework.

2 Data

2.1 Multilingual trainings

Multilingual models trained for the competition use a target language tag at the beginning of sentence to select the direction of the translation.

2.2 Transliteration

Indian languages use a variety of scripts. Using transliteration between scripts of similar languages may improve the quality of multilingual models as described in (Bawden et al., 2019) (Goyal and Sharma, 2019). The transliteration we applied was to replace Indian letters of various scripts to their equivalents in Devanagari script. We used indicNLP [2] library to perform the transliteration.

In our previous experiments with Indian languages we noticed an overall improvement of the quality for multi-indian models, so we used transliteration in all trainings. However, additional experiments on transliteration during the competition were not conclusive. The results for trainings on raw corpora, without transliteration were similar (see Table 1).

[1] github.com/marian-nmt/marian
[2] https://github.com/anoopkunchukuttan/indic_nlp_library

Proceedings of the 8th Workshop on Asian Translation, pages 224–232
Bangkok, Thailand (online), August 5-6, 2021.

2.3 Parallel Corpora Filtering

The base corpus for all trainings was the concatentaion of complete bilingual corpora provided by the organizers (further referenced as *bitext*) (11M lines in total). No filtering or preprocessing (but the transliteration) were performed on this corpus. The corpus included parallel data from: CVIT-PIB, PMIndia, IITB 3.0, JW, NLPC, UFAL EnTam, Uka Tarsadia, Wiki Titles, ALT, OpenSubtitles, Bibleuedin, MTEnglish2Odia, OdiEnCorp 2.0, TED, WikiMatrix. During the competition we performed several experiments to enrich/filter this parallel corpora:

- Inclusion of CCAligned corpus
- Removing *far from domain* sentence pairs like religious corpora
- Removing sentence pairs of low probability (according to e.g. sentence lengths, detected language etc.)
- Domain adaptation by fastText
- Domain adaptation by language model

None of these techniques applied on parallel corpora had led to quality improvement which is why we decided to continue with the basic non-filtered corpora as the base for future trainings.

2.4 Backtranslation

Backtranslation of monolingual corpora is a commonly used technique for improving machine translation. Especially for low-resource languages where only small bilingual corpora are available (Edunov et al., 2018). Training on backtranslations enriches the target language model, which improves the overall translation quality. The synthetic backtranslated corpus was joined with the original bilingual corpus for the trainings.

Using backtranslations of the full monolingual corpuses led to the improvement of results on translation on Indian to English directions by 1.2 BLEU on average. There was no improvement in the opposite directions. See Tables 5 and 6.

2.5 Domain adaptation

We enriched the parallel training corpora with backtranslated monolingual data selecting only sentences similar to PMI domain to increase the rate of in-domain data in the training corpus. We used two different techniques to select the in domain sentences for backtranslation. With these techniques we trained two separate families of MT models.

Domain adaptation by fastText (FT) - We applied the domain adaptation described in (Yu et al., 2020). Following the hints from the paper, we trained the fastText (Joulin et al., 2017) model using balanced corpus containing sentences from PMIndia labelled as in-domain and CCAligned sentences labelled as out-domain. Using the trained model we filtered the parallel as well as monolingual corpora.

Domain adaptation by language model (LM) As the second approach to select a subset of best PMI-like sentences from monolingual general-domain AI4Bharat (Kunchukuttan et al., 2020) corpora available for the task, we used the approach described in (Axelrod et al., 2011). For each of 10 Indian languages two RNN language models were constructed using Marian toolkit: in-domain trained with a particular part of PMI corpus and out-of-domain created using a similar number of lines from a mix of all other corpora available for that language respectively. All these models were regularized with exponential smoothing of 0.0001, dropout of 0.2 along with source and target word token dropout of 0.1. For the AI4Bharat mono corpus sentence ranking, we used a cross-entropy difference between scores of previously mentioned models as suggested in (Axelrod et al., 2011), normalized by the line length. Only sentences with a score above arbitrarily chosen threshold were selected for further processing. We noticed a significant influence of domain adaptation while selecting mono corpora used for backtranslation (see Table 3).

2.6 Multi-Agent Dual Learning

For some of trainings, we used the simplified version of Multi-Agent Dual Learning (MADL) (Wang et al., 2019), proposed in Kim et al. (2019), to generate additional training data from the parallel corpus. We generated n-best translations of both the source and the target sides of the parallel data, with strong ensembles of, respectively, the forward and the backward models. Next, we picked the best translation from among n candidates w.r.t. the sentence-level BLEU score. Thanks to these steps, we tripled the number of sentences by combining three types of datasets:

1. original source – original target,
2. original source – synthetic target,
3. synthetic source – original target,

where the synthetic target is the translation of the original source with the forward model, and the synthetic source is the translation of the original target with the backward model.

2.7 Postprocessing

In comparison to our competitors we noticed significantly weaker performance on the En-Or direction. After the analysis we found out that the generated corpora contain sequences of characters (U+0B2F-U+0B3C, U+0B5F) not present in the devset corpora. Replacing these sequences with sequence (U+0B5F-U+0B3E) gave a significant improvement for En-Or of about +4 BLEU.

3 NMT System Overview

All of our systems are trained with the Marian NMT[3] (Junczys-Dowmunt et al., 2018) framework.

3.1 Baseline systems for preliminary experiments

First experiments were performed with transformer models (Vaswani et al., 2017), which we will now refer to as *transformer-base*. The only difference is that we used 8 encoder layers and 4 decoder layers instead of default configuration 6-6. The model has default embedding dimension of 512 and a feed-forward layer dimension of 2048.

We also used layer normalization (Ba et al., 2016) and tied the weights of the target-side embedding and the transpose of the output weight matrix, as well as source- and target-side embeddings (Press and Wolf, 2017). Optimizer delay was used to simulate batches of size up to 200GB, Adam (Kingma and Ba, 2017) was used as an optimizer, with a learning rate of 0.0003 and linear warm-up for the initial $48,000$ updates with subsequent inverted squared decay. No dropout was applied.

3.2 Final configuration

After the first experiments further trainings were performed on a *transformer-big* model. It has bigger dimensions than the *transformer-base*: an embedding dimension of 1024 and a feed-forward layer dimension of 4096. The *transformer-big* trainings were regularized with a dropout between transformer layers of 0.1 and a label smoothing of 0.1 unlike the *transformer-base* which was trained without a dropout.

[3] github.com/marian-nmt/marian

Parallel	En-In	In-En
bitext	18.03	31.41
CCAligned	6.82	12.15
PMIndia	5.59	11.94
bitext+CC	17.62	30.56
bitext, no religious	15.33	29.02
bitext, filtered FT	17.84	29.38
bitext, most likely	17.98	31.00
bitext, no transliteration	18.36	31.27
With backtranslation		
bitext+BT filtered LM	18.22	31.38
bitext+BT filtered FT	18.71	32.77
bitext+CC+BT flitered FT	18.21	30.64
MADL		
MADL	18.87	31.94
MADL+BT filtered FT	18.83	33.25

Table 1: Average BLEU for preliminary trainings (4.1) on different corpora.

4 Trainings

4.1 Preliminary trainings

During preliminary trainings, we tested which techniques of filtering/backtranslation/MADL work best for the task. Preliminary trainings were performed for all 20 directions on a single *transformer-base* model with no dropout.

There was no clear answer, which of the techniques work best. Generally, adding CCAligned corpus worsened the results. Training only on a big CCAligned corpus (15M lines) gave similar results to training on small PMIndia corpus (300k lines). For further trainings we decided to use the most promising techniques: filtered backtranslation (both methods fastText and Language Model) and MADL.

The preliminary training for one *transformer-base* model lasted 50 hours on two V100 GPUs - 13 epochs. A summary of the preliminary results are gathered in Table 1

4.2 Pretraining with backtranslations

For the final trainings we prepared various corpora with backtranslations filtered with a domain-transfer. We applied two methods of domain-

fastText filtering	Source	Selected
backtranslations	400M	86M
bitext filtered FT	11M	1,5M
CCAligned	15M	400k
PMIndia	300k	300k
bitext full	11M	11M
Language Model filtering	**Source**	**Selected**
backtranslations	400M	58M
bitext full	11M	11M
bitext distilled forward	11M	11M
bitext distilled backward	11M	11M

Table 2: Components of mixed corpora used for pretrainings with backtranslation (4.2) using fastText filtering and language model filtering of monolingual corpora.

	BLEU		Improvement	
No filtering	**2In**	**2En**	**2In**	**2En**
Bitext only	18.81	31.80		
Full BT	18.77	33.02	-0.04	1.22
LM filtering				
Filtered BT	20.06	35.43	1.25	3.63
Tuned Bitext	21.03	36.95	0.97	1.52
FT PMIndia	21.39	37.26	0.36	0.31
fastText filtering				
Filtered BT	19.77	36.62	0.96	4.86
Tuned BT-PMI	21.01	37.64	1.24	1.02
Tuned Bitext	21.31	38.47	1.54	1.85
FT PMIndia	**21.91**	**38.72**	0.60	0.25
FT BT-PMI	21.81	38.42	0.50	-0.05
FT MADL		38.67		0.20

Table 3: Comparison of domain-adaptation techniques - Average BLEU over 10 directions for subsequent stages of final training: pretraining with backtranslation, tuning with bitext, tuning with mono PMIndia backtranslated, finetuning with bitext PMIndia, finetuning with backtranslated mono PMIndia, finetuning with MADL.

transfer described in previous sections: fastText and language model. Trainings were performed on separate *transformer-big* models. One *many-to-one* model for 10 directions to-English and second *one-to-many* for 10 directions from-English.

The whole pretraining for one *transformer-big* model lasted 200 hours on four V100 GPUs - 8 epochs. Further tunings took additional 20 hours of processing.

4.3 Tuning with bitext

The best two pretrained models with domain-transfer (LM filtered and FT filtered) were the baselines to start the tuning with the parallel corpora. During the bitext tuning we used all bilingual data provided by organizers except CCAligned corpus - 11M sentences in total. Tuning of baselines with the original parallel corpora improved the average BLEU of pretrained models by 0.97-1.85 BLEU (see Table 3)

4.4 Finetuning with PMIndia

We performed several attempts to finetune the final results with different corpora:

1. PMIndia parallel corpus (300k lines)
2. Baktranslated PMIndia mono corpus (1,1M lines)
3. MADL on PMIndia parallel corpus (3 * 300k lines)

First of these attempts, finetuning with bilingual PMIndia, gave the best improvement of final result - 0.25-0.6 BLEU on average. All 3 finetuned models were taken into process of mixing the best ensemble.

4.5 Ensembling

To further boost the translation quality, we used ensembles of models during decoding. Two separate ensembles were formed and tuned, one for transliterated Indian to English, the other in the opposite direction. Each ensemble consisted of: a number of Neural Translation Models, derived from various stages of training and model tuning – up to as much as 9 NMT were used during weight-optimization; and a single Neural Language Model, either English or common Indian (based on all languages, transliterated into Hindi), depending on the direction.

The tuning of ensemble weights was performed on the Development set and consisted of the following stages:

- Expectation-Maximization of posterior emission probability for a mixture of models(Kneser and Steinbiss, 1993), based on NMT log-scores of Development sentence-pairs, obtained using `marian-score`; this procedure, as well as being fast due to *not* requiring actual decoding, also worked well in practice, despite being based on interpolation

Set	Indian-En		En-Indian	
Technique	Dev	Test	Dev	Test
Best sng	41.39	38.52	22.32	20.50
Unif w/o LM	42.33	39.6	22.57	20.79
Unif. w/ LM	40.69	37.88	21.51	20.01
Expert sel.	42.11	39.24	22.62	20.94
E-M*	42.35	39.71	22.64	20.99
+ ind. wgts	42.49	39.65	22.74	20.99
+ norm-fact.	42.50	39.58	n/a	n/a

Table 4: BLEU scores for different techniques of determining ensemble weights.
* Expectation-Maximization of likelihoods optimized weights of translation models only; Language Model was then added with small arbitrary weight of ca. 0.3%, and the presented scores were obtained using such an ensemble.

in the linear probability domain, as opposed to log-domain interpolation used in Marian;

- tuning single weights of the ensemble (bisectioning procedure, performed for a limited number of iterations; weights were tuned in the arbitrary order), based on BLEU scores of translated Development set (before normalization and tokenization);
- (only for Indian-to-English) a sweep of normalization-factor, also on BLEU. [4]

Individual tuning for target languages of English-to-Indian directions was originally planned, but wasn't eventually used for submission, mostly due to lack of time, however visual inspection of the partial results also showed that some weights varied wildly, so devset over-fitting could be suspected at this point; normalization-factor optimization was planned to be performed after the aforementioned optimization, so consequently it was also skipped for English-to-Indian directions. Post-submission tests showed an average improvement of ca. 0.2 BLEU, when using tuning for individual Indian target languages, but the gain was strongly dominated by the improvement on a single direction (En→Hi).

We experimented with several beam sizes increasing it up to 40. For the final submission we chose the size of 16. The larger beam gave little or no improvement at a cost of slowing down the decoding. For very large ensembles of 10 big models the decoding of the whole devset for 10 directions (10k lines) lasts about 25 minutes on a single V100 GPU.

Table 4 presents the impact of tuning on BLEU scores on both devset and testset, in relation to a few manually selected setups, namely best-single-model, uniform and expert-selected "50-25-25%" ensemble. The final weight selection improved translation of the Indian-to-En directions by ca. 0.5 BLEU, compared to the expert ensemble or ca. 1.2 BLEU compared to best single model; on En-to-Indian directions, the improvement was <0.1 BLEU or ca. 0.5 BLEU, respectively. The results on the testset differ slightly from our final submissions as, during the ensemble tuning, we used simplified BLEU calculations algorithm (before normalization and tokenization)

5 Final Results

The detailed results of each stage of the best branch of trainings are gathered in Tables 5 and 6. The ensemble values are the submission evaluation results provided by the organizers.

Tables 7 and 8 contain the results of the models submitted by SRPOL compared with best results of competitors. The tables present values provided by WAT2021 organizers, calculated by 3 different metrics: BLEU (Papineni et al., 2002), RIBES (Isozaki et al., 2010), AMFM (Banchs and Li, 2011)

Figure 1 shows the results of the human evaluation. The figure presents the values provided by WAT2021 organizers showing significant advance over the competitors. Especially amount of bad translations (scored 1-2) has been significantly reduced.

5.1 English → Indian

Application of all techniques for En→In directions gave the overall improvement of 3.6 BLEU from baseline average 18.8 to final 22.4 BLEU. Adding non-filtered backtranslations gave no improvement, probably because general Indian monocorpus is too different from specific language used in PMIndia. However, after domain adaptation of the training corpus we gained improvement of 1 BLEU. Most of the improvement was gained by finetuning on parallel corpora (1.5 BLEU) and PMI corpora (0.6 BLEU). Final ensembling process gave the average improvement of 0.5 BLEU.

[4] Translation score of each hypothesis is divided by $length^{factor}$, this value is then used to select the final translation, default is 1.

Stage	Bn	Gu	Hi	Kn	Ml	Mr	Or	Pa	Ta	Te	AVG	Boost
Baseline - bitext	13.1	23.7	35.8	15.8	12.2	16.7	17.0	29.8	12.0	11.9	18.81	
Backtranslations	12.5	23.5	36.1	16.6	12.4	17.1	17.0	29.8	11.8	11.0	18.77	-0.04
Domain adapt.	13.4	23.8	36.8	17.2	13.8	18.7	18.3	30.6	12.9	12.2	19.77	1.00
Tuning bitext	14.6	25.9	38.1	19.5	14.9	19.6	19.5	32.3	13.6	14.9	21.31	1.54
Tuning PMIndia	15.5	27.2	38.1	20.8	15.1	19.8	19.1	32.9	13.7	16.8	21.91	0.60
Ensemble	16.0	27.8	38.7	21.3	15.5	20.4	19.9	33.4	14.2	16.9	22.40	0.49

Table 5: Final results - BLEU for 10 directions from-English in subsequent stages of final training

Stage	Bn	Gu	Hi	Kn	Ml	Mr	Or	Pa	Ta	Te	AVG	Boost
Baseline - bitext	25.2	36.4	39.9	31.0	29.5	29.8	30.2	38.0	28.5	29.5	31.80	
Backtranslations	25.3	37.8	40.6	33.6	30.8	30.6	31.8	39.2	29.3	31.2	33.02	1.22
Domain adapt.	29.4	40.6	44.5	36.9	35.1	33.6	35.2	42.7	33.0	35.0	36.62	3.60
Tuning bitext	31.1	42.7	45.3	38.9	37.2	35.2	36.2	44.8	34.9	38.3	38.47	1.85
Tuning PMIndia	31.8	43.3	45.6	39.1	37.1	35.7	36.2	44.8	35.0	38.6	38.72	0.25
Ensemble	31.9	44.0	46.9	40.3	38.4	36.6	37.1	46.4	36.1	39.8	39.75	1.03

Table 6: Final results - BLEU for 10 directions to-English in subsequent stages of final training

5.2 Indian → English

Application of all techniques for In→En directions gave the overall improvement of 8 BLEU from baseline average 31.8 to final 39.8 BLEU. Adding non-filtered backtranslations gave 1.2 BLEU improvement but most of the improvement had been gained by domain adaptation which gave surprisingly high improvement of 3.6 BLEU. Further improvement was gained by finetuning on parallel corpora (1.9 BLEU) and PMI corpora (0.3 BLEU). The final ensembling process gave additional improvement of 1.0 BLEU.

6 Conclusions

We presented an effective approach to low-resource training consisting of pretraining on backtranslations and tuning on parallel corpora. We successfully applied domain-adaptation techniques which significantly improved translation quality measured by BLEU. We presented an effective approach for finding best hyperparameters for the ensembling number of single translation models.

We applied transliteration, but the final results did not confirm that this approach is effective, at least for that particular task.

We tried several filtering techniques for parallel corpora but the results showed no improvement. This may be a confirmation that the parallel corpora provided by the competition organizers are of high quality which is hard to improve.

Probably for the same reason domain-adaptation on parallel corpora didn't improve the results. However domain-adaptation worked surprisingly well for monolingual corpora.

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. 2016. Layer normalization.

Rafael E. Banchs and Haizhou Li. 2011. AM-FM: A semantic framework for translation quality assessment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 153–158, Portland, Oregon, USA. Association for Computational Linguistics.

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.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A brief survey of multilingual neural machine translation.

Model	Bn	Gu	Hi	Kn	Ml	Mr	Or	Pa	Ta	Te	AVG
BLEU											
Baseline	12.03	22.99	35.25	14.72	11.93	16.07	12.33	28.65	11.44	10.65	17.61
Competitor	14.73	26.97	38.25	19.57	12.79	19.48	**20.15**	33.35	**14.43**	15.61	21.53
Best single	15.58	27.31	38.04	20.91	15.43	19.93	19.15	32.88	13.89	16.82	21.99
Ensemble	**15.97**	**27.80**	**38.65**	**21.30**	**15.49**	**20.42**	19.94	**33.43**	14.15	**16.85**	**22.40**
RIBES											
Baseline	0.7072	0.8020	0.8438	0.7281	0.6874	0.7388	0.7146	0.8203	0.6971	0.6924	0.7432
Competitor	0.7242	0.8202	0.8542	0.7601	0.7074	0.7600	0.7503	**0.8376**	0.7215	0.7284	0.7664
Best single	0.7328	0.8223	0.8525	0.7701	0.7341	0.7669	0.7497	0.8355	0.7288	0.7345	0.7727
Ensemble	**0.7336**	**0.8249**	**0.8559**	**0.7712**	**0.7369**	**0.7718**	**0.7511**	0.8375	**0.7307**	**0.7398**	**0.7753**
AMFM											
Baseline	0.7675	0.8166	0.8224	0.8091	0.7986	0.8050	0.7146	0.7733	0.7957	0.7633	0.7866
Competitor	**0.7796**	0.8201	0.8228	0.8178	0.8053	0.8115	0.7699	0.8137	**0.8029**	0.7898	**0.8033**
Best single	0.7723	0.8199	0.8224	0.8213	0.8080	**0.8108**	0.7715	0.8132	0.7994	**0.7930**	0.8032
Ensemble	0.7710	**0.8212**	**0.8246**	**0.8219**	**0.8081**	0.8097	**0.7718**	**0.8141**	0.7988	0.7911	0.8032

Table 7: Official results of translations from-English by 3 metrics for submitted results of: baseline model, best competitor's result, submitted single SRPOL's model and submitted best SRPOL's ensemble

Model	Bn	Gu	Hi	Kn	Ml	Mr	Or	Pa	Ta	Te	AVG
BLEU											
Baseline	25.39	35.86	39.49	30.67	28.69	29.10	30.07	37.61	28.01	29.05	31.39
Competitor	29.96	39.39	43.23	35.46	33.21	34.02	34.11	41.24	31.94	35.44	35.80
Best single	31.82	42.87	45.61	39.01	37.04	35.68	36.04	44.87	35.06	38.57	38.66
Ensemble	**31.87**	**43.98**	**46.93**	**40.34**	**38.38**	**36.64**	**37.06**	**46.39**	**36.13**	**39.80**	**39.75**
RIBES											
Baseline	0.7649	0.8186	0.8448	0.7984	0.7927	0.7879	0.7895	0.8335	0.7881	0.7803	0.7999
Competitor	0.7983	0.8394	0.8591	0.8209	0.8132	0.8103	0.8017	0.8495	0.8070	0.8168	0.8216
Best single	0.8001	0.8497	0.8677	0.8373	0.8304	0.8212	0.8128	0.8614	0.8160	0.8315	0.8328
Ensemble	**0.8005**	**0.8533**	**0.8729**	**0.8405**	**0.8354**	**0.8248**	**0.8170**	**0.8658**	**0.8223**	**0.8364**	**0.8369**
AMFM											
Baseline	0.7699	0.8129	0.8250	0.7927	0.7936	0.7916	0.7940	0.8151	0.7884	0.7872	0.7970
Competitor	0.7786	0.8207	0.8345	0.8097	0.8068	0.7958	0.8082	0.8235	0.7961	0.8040	0.8078
Best single	**0.7924**	0.8331	0.8435	0.8204	0.8207	0.8103	0.8149	0.8364	0.8036	0.8204	0.8196
Ensemble	0.7897	**0.8358**	**0.8471**	**0.8237**	**0.8230**	**0.8123**	**0.8173**	**0.8416**	**0.8065**	**0.8209**	**0.8218**

Table 8: Official results of translations to-English by 3 metrics for submitted results of: baseline model, best competitor's result, submitted single SRPOL's model and submitted best SRPOL's ensemble

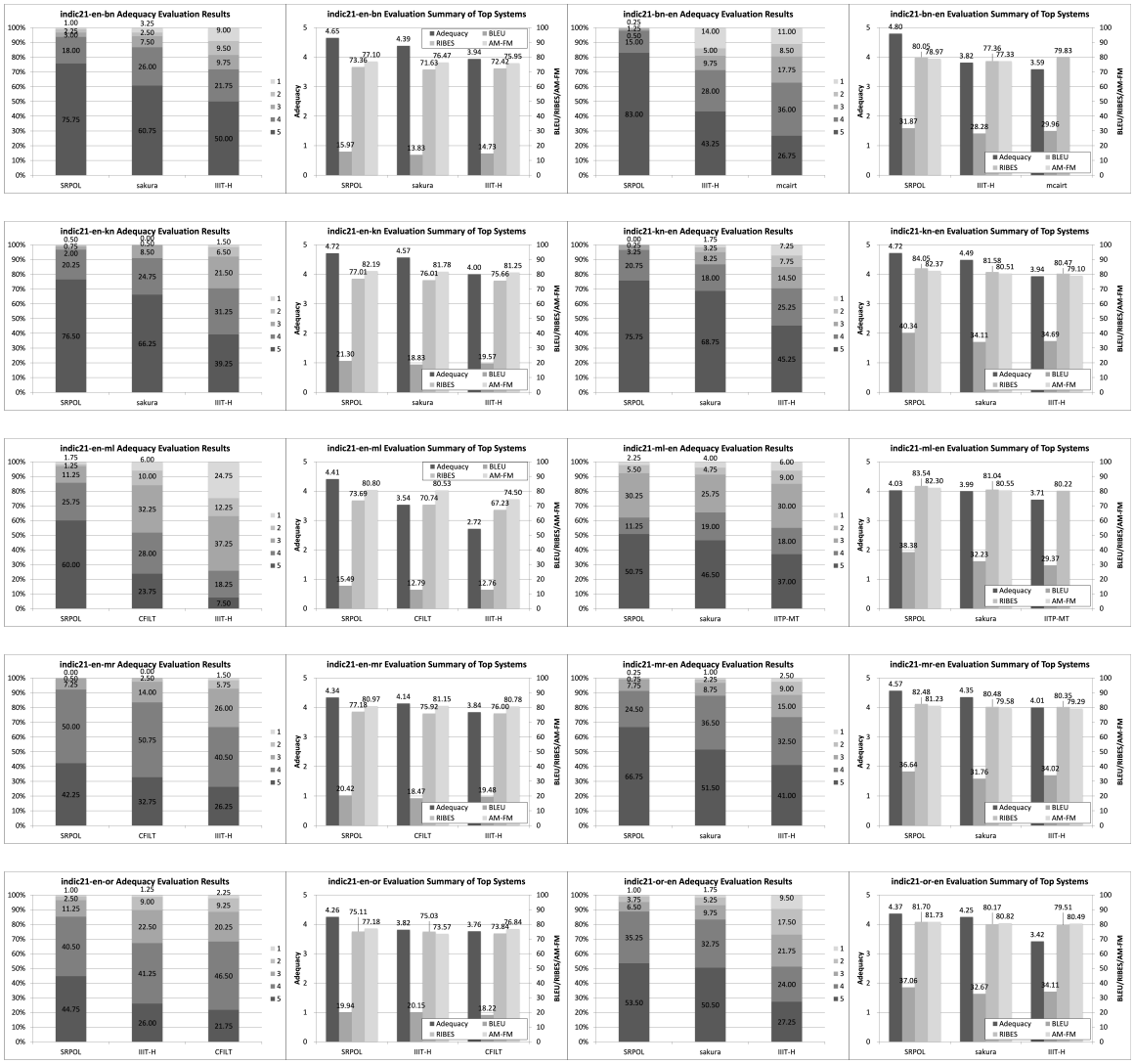

Figure 1: Summary results for all 5 manually evaluated languages - Bengali, Kannada, Malayalam, Marathi, Oriya

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500, Brussels, Belgium. Association for Computational Linguistics.

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, Florence, Italy. Association for Computational Linguistics.

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952, Cambridge, MA. Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431. Association for Computational Linguistics.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Young Jin Kim, Marcin Junczys-Dowmunt, Hany Hassan, Alham Fikri Aji, Kenneth Heafield, Roman Grundkiewicz, and Nikolay Bogoychev. 2019. From research to production and back: Ludicrously fast neural machine translation. In *Proceedings of the 3rd Workshop on Neural Generation and Translation*, pages 280–288, Hong Kong. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Reinhard Kneser and Volker Steinbiss. 1993. On the dynamic adaptation of stochastic language models. In *1993 IEEE International Conference on Acoustics, Speech, and Signal Processing*, volume 2, pages 586–589 vol.2.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Gokul N. C., Avik Bhattacharyya, Mitesh M. Khapra, and Pratyush Kumar. 2020. Ai4bharat-indicnlp corpus: Monolingual corpora and word embeddings for indic languages.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, Chenhui Chu, Akiko Eriguchi, Kaori Abe, and Sadao Oda, Yusuke Kurohashi. 2021. Overview of the 8th workshop on Asian translation. In *Proceedings of the 8th Workshop on Asian Translation*, Bangkok, Thailand. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models.

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

Yiren Wang, Yingce Xia, Tianyu He, Fei Tian, Tao Qin, ChengXiang Zhai, and Tie-Yan Liu. 2019. Multi-agent dual learning. In *International Conference on Learning Representations*.

Zhengzhe Yu, Zhanglin Wu, Xiaoyu Chen, Daimeng Wei, Hengchao Shang, Jiaxin Guo, Zongyao Li, Minghan Wang, Liangyou Li, Lizhi Lei, Hao Yang, and Ying Qin. 2020. HW-TSC's participation in the WAT 2020 indic languages multilingual task. In *Proceedings of the 7th Workshop on Asian Translation*, pages 92–97, Suzhou, China. Association for Computational Linguistics.