IDIAP RESEARCH REPORT

NLPHUT'S PARTICIPATION AT WAT2021

Shantipriya Parida[a] Subhadarshi Panda
Ketan Kotwal Amulya Ratna Dash
Satya Ranjan Dash Yashvardhan Sharma
Petr Motlicek Ondrej Bojar

Idiap-RR-10-2021

JULY 2021

[a]Idiap Research Institute

Centre du Parc, Centre du Parc, Rue Marconi 19, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

NLPHut's Participation at WAT2021

Shantipriya Parida*, Subhadarshi Panda†, Ketan Kotwal*, Amulya Ratna Dash♣, Satya Ranjan Dash♠, Yashvardhan Sharma♣, Petr Motlicek*, Ondřej Bojar◇

*Idiap Research Institute, Martigny, Switzerland
{firstname.lastname}@idiap.ch
†Graduate Center, City University of New York, USA
spanda@gradcenter.cuny.edu
♣Birla Institute of Technology and Science, Pilani, India
{p20200105,yash}@pilani.bits-pilani.ac.in
♠KIIT University, Bhubaneswar, India
sdashfca@kiit.ac.in
◇Charles University, MFF, ÚFAL, Prague, Czech Republic
bojar@ufal.mff.cuni.cz

Abstract

This paper provides the description of shared tasks to the WAT 2021 by our team "NLPHut". We have participated in the English→Hindi Multimodal translation task, English→Malayalam Multimodal translation task, and Indic Multilingual translation task. We have used the state-of-the-art *Transformer* model with language tags in different settings for the translation task and proposed a novel "region-specific" caption generation approach using a combination of image CNN and LSTM for the Hindi and Malayalam image captioning. Our submission tops in English→Malayalam Multimodal translation task (text-only translation, and Malayalam caption), and ranks second-best in English→Hindi Multimodal translation task (text-only translation, and Hindi caption). Our submissions have also performed well in the Indic Multilingual translation tasks.

1 Introduction

Machine translation (MT) is considered to be one of the most successful applications of natural language processing (NLP)[1]. It has significantly evolved especially in terms of the accuracy of its output. Though MT performance reached near to human level for several language pairs (see e.g. Popel et al., 2020), it remains challenging for low resource languages or translation effectively utilizing other modalities (e.g. image, Parida et al., 2020).

The Workshop on Asian Translation (WAT) is an open evaluation campaign focusing on Asian languages since 2013 (Nakazawa et al., 2020). In WAT2021 (Nakazawa et al., 2021) Multimodal track, a new Indian language *Malayalam* was introduced for English→Malayalam text, multimodal translation, and Malayalam image captioning task.[2] This year, the MultiIndic[3] task covers 10 Indic languages and English.

In this system description paper, we explain our approach for the tasks (including the subtasks) we participated in:

Task 1: English→Hindi (EN-HI) Multimodal Translation
- EN-HI text-only translation
- Hindi-only image captioning

Task 2: English→Malayalam (EN-ML) Multimodal Translation
- EN-ML text-only translation
- Malayalam-only image captioning

Task 3: Indic Multilingual translation task.

Section 2 describes the datasets used in our experiment. Section 3 presents the model and experimental setups used in our approach. Section 4 provides the official evaluation results of WAT2021[4] followed by the conclusion in Section 5.

[1] https://morioh.com/p/d596d2d4444d

[2] https://ufal.mff.cuni.cz/malayalam-visual-genome/wat2021-english-malayalam-multi

[3] http://lotus.kuee.kyoto-u.ac.jp/WAT/indic-multilingual/

[4] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2021/index.html

2 Dataset

We have used the official datasets provided by the WAT2021 organizers for the tasks.

Task 1: English→Hindi Multimodal Translation For this task, the organizers provided HindiVisualGenome 1.1 (Parida et al., 2019)[5] dataset (HVG for short). The training part consists of 29k English and Hindi short captions of rectangular areas in photos of various scenes and it is complemented by three test sets: development (D-Test), evaluation (E-Test) and challenge test set (C-Test). Our WAT submissions were for E-Test (denoted "EV" in WAT official tables) and C-Test (denoted "CH" in WAT tables). Additionally, we used the IITB Corpus[6] which is supposedly the largest publicly available English-Hindi parallel corpus (Kunchukuttan et al., 2017). This corpus contains 1.59 million parallel segments and it was found very effective for English-Hindi translation (Parida and Bojar, 2018). The statistics of the datasets are shown in Table 1.

		Tokens		
Set	Sentences	English	Hindi	Malayalam
Train	28930	143164	145448	107126
D-Test	998	4922	4978	3619
E-Test	1595	7853	7852	5689
C-Test	1400	8186	8639	6044
IITB Train	1.5 M	20.6 M	22.1 M	–

Table 1: Statistics of our data used in the English→Hindi and English→Malayalam Multimodal task: the number of sentences and tokens.

Task 2: English→Malayalam Multimodal Translation For this task, the organizers provided MalayalamVisualGenome 1.0 dataset[7] (MVG for short). MVG is an extension of the HVG dataset for supporting Malayalam, which belongs to the Dravidian language family (Kumar et al., 2017). The dataset size and images are the same as HVG. While HVG contains bilingual (English and Hindi) segments, MVG contains bilingual (English and Malayalam) segments, with the English shared across HVG and MVG, see Table 1.

[5] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3267

[6] http://www.cfilt.iitb.ac.in/iitb_parallel/

[7] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-3533

Task 3: Indic Multilingual Translation For this task, the organizers provided a training corpus that comprises in total 11 million sentence pairs collected from several corpora. The evaluation (dev and test set) contain filtered data of the PMIndia dataset (Haddow and Kirefu, 2020).[8] We have not used any additional resources in this task. The statistics of the dataset are shown in Table 2.

3 Experimental Details

This section describes the experimental details of the tasks we participated in.

3.1 EN-HI and EN-ML text-only translation

For the HVG text-only translation track, we train a Transformer model (Vaswani et al., 2017) using the concatenation of IIT-B training data and HVG training data (see Table 1). Similar to the two-phase approach outlined in Section 3.3, we continue the training using only the HVG training data to obtain the final checkpoint. For the MVG text-only translation track, we train a Transformer model using only the MVG training data.

For both EN-HI and EN-ML translation, we trained SentencePiece subword units (Kudo and Richardson, 2018) setting maximum vocabulary size to 8k. The vocabulary was learned jointly on the source and target sentences of HVG and IIT-B for EN-HI and of MVG for EN-ML. The number of encoder and decoder layers was set to 3 each; while the number of heads was set to 8. We have set the hidden size to 128, along with the dropout value of 0.1. We initialized the model parameters using Xavier initialization (Glorot and Bengio, 2010) and used the Adam optimizer (Kingma and Ba, 2014) with a learning rate of $5e-4$ for optimizing model parameters. Gradient clipping was used to clip gradients greater than 1. The training was stopped when the development loss did not improve for 5 consecutive epochs. While EN-HI training using concatenated IIT-B + HVG data and the subsequent training using only HVG data, we used the same HVG dev set for determining early stopping. For generating translations, we used greedy decoding and generated tokens autore-

[8] http://data.statmt.org/pmindia/

Language pair	en-bn	en-hi	en-gu	en-ml	en-mr	en-ta	en-te	en-pa	en-or	en-kn
Train (ALL)	1756197	3534387	518015	1204503	781872	1499441	686626	518508	252160	396865
Train (PMI)	23306	50349	41578	26916	28974	32638	33380	28294	31966	28901
Dev	1000									
Test	2390									

Table 2: Statistics of the data used for Indic multilingual translation.

gressively till the end-of-sentence token was generated or the maximum translation length was reached, which was set to 100.

We show the training and development perplexities for EN-HI and EN-ML translations during training in Figure 4b. The dev perplexity for EN-HI translation is lower in the beginning (after epoch 1) because the model is trained using more training samples (IIT-B + HVG) in comparison to EN-ML. Overall, EN-HI training takes around twice as much time as EN-ML training, again due to the involvement of the bigger IIT-B training data. The drop in perplexity midway for EN-HI is because of the change of training data from IIT-B + HVG to only HVG after the first phase of the training converges.

Upon evaluating the translations using the development set, we obtained the following scores for Hindi translations. The BLEU score was 46.7 upon using HVG + IIT-B training data. In comparison, we observed that the BLEU score was 39.9 upon using only the HVG training data (without IIT-B training data). For Malayalam translations, the BLEU score on the development set was 31.3. BLEU scores were computed using sacreBLEU (Post, 2018).

3.2 Image Caption Generation

This task in WAT 2021 is formulated as generating a caption in Hindi and Malayalam for a specific region in the given image. Most existing research in the area of image captioning refers to generating a textual description for the entire image (Yang and Okazaki, 2020; Yang et al., 2017; Lindh et al., 2018; Staniūtė and Šešok, 2019; Miyazaki and Shimizu, 2016; Wu et al., 2017). However, a naive approach of using only a specified region (as defined by the rectangular bounding box) as an input to the generic image caption generation system often does not yield meaningful results. When a small region of the image with few objects is considered for captioning, it lacks the context

English Text: The snow is white. Hindi Text: बर्फ सफेद है

Malayalam Text: മഞ്ഞ് വെളുത്തതാണ് Gloss: Snow is white

Figure 1: Sample image with specific region and its description for caption generation. Image taken from Hindi Visual Genome (HVG) and Malayalam Visual Genome (MVG) (Parida et al., 2019)

(*i.e.,* overall understanding) around the region that can essentially be captured from the entire image as shown in Figure 1. It is challenging to generate the caption "snow" only considering the specific region (red bounding box).

We propose a region-specific image captioning method through the fusion of encoded features of the region as well as that of the complete image. Our proposed model for this task consists of three modules – an encoder, fusion, and decoder – as shown in Figure 2.

Image Encoder: To textually describe an image or a region within, it first needs to be encoded into high-level complex features that capture its visual attributes. Several image captioning works (Yang and Okazaki, 2020; Yang et al., 2017; Lindh et al., 2018; Staniūtė and Šešok, 2019; Miyazaki and Shimizu, 2016; Wu et al., 2017) have demonstrated that the outputs of final or pre-final convolutional (conv) layers of deep CNNs are excellent features for the aforementioned objective. Along with features of the entire image, we propose to extract the features of the subregion as well using the same set of outputs of the conv layer. Let $\mathbf{F} \in \mathbb{R}^{MNC}$ be the features of the final conv

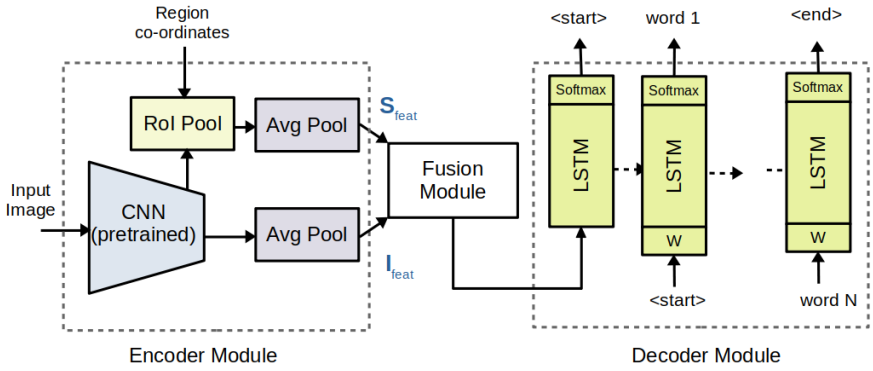


Figure 2: Architecture of the proposed model for region-specific image caption generator. The Encoder module consists of a pre-trained image CNN as feature extractor, while an LSTM-based decoder generates captions. Both modules are connected by a Fusion module.

layer of a pre-trained image CNN where C represents the number of channels or maps, and M, N are the spatial dimensions of each feature map. From the dimensions of the input image and the values of M, N, we compute the spatial scaling factor. Through this factor and nominal interpolation, we obtain a corresponding location of the subregion in the conv layer, say with dimensionality (m, n). This subset, $\mathbf{F}_s \in \mathbb{R}^{mnC}$, predominantly consists of features from the subregion. The subset $\mathbf{F}_s$ is obtained through the region of interest (RoI) pooling (Girshick, 2015). We do not modify the channel dimensions of $\mathbf{F}_s$. The final features, thus obtained, are linearized to form a single column vector. We denote the region-subset features as S_{feat}. The features of the complete image are nothing but $\mathbf{F}$. We apply spatial pooling on this feature set to reduce their dimensionality, and obtain the linearized vector of full-image features denoted as I_{feat}.

Fusion Module: The region-level features capture details of the region (objects) to be described; whereas image-level features provide an overall context. To generate meaningful captions for a region of the image, we consider the features of the region S_{feat} along with the features of the entire image I_{feat}. This combining of feature vectors is crucial in generating descriptions for the region. In this work, we propose to conduct fusion through the concatenation of weighted features from the region and those from the entire image for region-specific caption generation. The fused feature, $\mathbf{f}$, can be represented as $\mathbf{f} = [\alpha\, S_{\text{feat}}; (1-\alpha)\, I_{\text{feat}}]$, where α is the weightage parameter in [0.50, 1] indicating relative importance provided to region-features S_{feat} over the features of the whole image. For $\alpha = 0.66$, the region-level features are weighted twice as high as the entire image-level features. The weighing of a feature vector scales the magnitude of the corresponding vector without altering its orientation. Unlike the fusion mechanisms based on weighted addition, we do not modify the complex information captured by the features (except for scale); however, its relative importance with respect to the other set of features is adjusted for better caption generation. The fused feature $\mathbf{f}$ with the dimensionality of the sum of both feature vectors are then fed to the LSTM-based decoder.

LSTM Decoder: In the proposed approach, the encoder module is not trainable, it only extracts the image features however the LSTM decoder is trainable. We used LSTM decoder using the image features for caption generation using greedy search approach (Soh). We used the cross-entropy loss during decoding (Yu et al., 2019).

3.3 Indic Multilingual Translation

Sharing parameters across multiple languages, particularly low-resource Indic languages, results in gains in translation performance (Dabre et al., 2020). Motivated by this finding, we train neural MT models with shared parameters across multiple languages for the Indic multilingual translation task. We additionally apply transfer learning where we train a neural MT model in two phases (Kocmi and Bojar, 2018). The first phase consists of

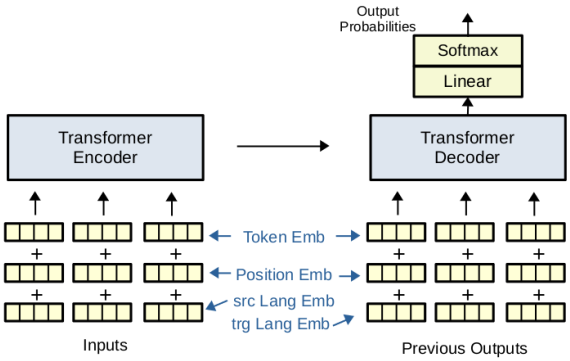


Figure 3: Architecture for Indic Multilingual translation. We show here the setup in which both the source and the target language tags are used.

training a multilingual translation model on training pairs drawn from one of the following options: (a) any Indic language from the dataset as the source and corresponding English target; (b) English as the source and any corresponding Indic language as the target; and (c) combination of (a) and (b), that is, the model is trained to enable translation from any Indic language to English and also English to any Indic language. The second phase involves fine-tuning of the model at the end of phase 1 using pairs from a single language pair. For phase 1, we used the PMI dataset for all the languages combined; whereas, for phase 2, we used either only the PMI portion or all the bilingual data available for the desired language pair. In Table 2, the training data sizes are denoted as *Train (PMI)* for phase 1 of training.

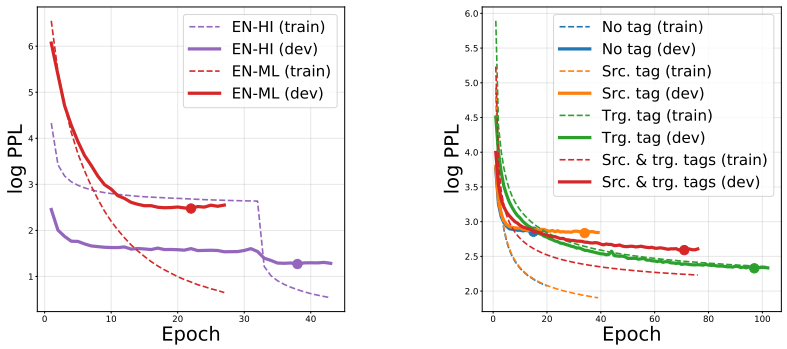
To support multilinguality (*i.e.,* going beyond a bilingual translation setup), we have to either fix the target language (many-to-one setup) or provide a language tag for controlling the generation process. We highlight below the four setups to achieve this:

Many-to-one setup with no tag In this setup, we use a transformer model (Vaswani et al., 2017) without any architectural modification that would enable the model to explicitly distinguish between languages. In phase 1 of the training process, we concatenate across all Indic languages the pairs drawn from an Indic language as the source and the corresponding English target and use the resulting data for training.

Many-to-one setup with source language tag We use a transformer model where the source language tag explicitly informs the model about the language of the source sentence as in Lample and Conneau (2019). We provide the language information at every position by representing each source token as the sum of token embedding, positional embedding, and language embedding; which is then fed to the encoder (see Figure 3 for the inputs to the encoder). The training data for phase 1 of the training process is the same as in the previous setup.

One-to-many setup with target language tag This setup is based on a transformer model where the target language embedding is injected to the decoder at every step and it explicitly informs the model about the desired language of the target sentence (Lample and Conneau, 2019). In this setup, the source is always in English. Similar to the previous setup, we represent each target token as the sum of token embeddings, positional embedding, and language embedding. Figure 3 shows the inputs to the decoder. In phase 1 of the training process, we concatenate across all Indic languages the pairs drawn from English as the source and the corresponding Indic language target and use the resulting data for training.

Many-to-many setup with both the source and target language tags In this setup, we use a transformer model where both the encoder and decoder are informed about the source and target languages explicitly through language embedding at every token (Lample and Conneau, 2019). For instance, the same model can be used for *hi-en* translation and also for *en-hi* translation. As shown in the architecture in Figure 3, the source token representation is computed as the sum of the token embedding, positional embedding, and source language embedding. Similarly, the target token representation is computed as the sum of the token embedding, positional embedding, and target language embedding. The source and the target token representations are provided to the encoder and decoder, respectively. The rest of the modules in the transformer model architecture are same as in Vaswani et al. (2017). The training



(a) (b)

Figure 4: Training and development perplexity for: (a) EN-HI and EN-ML translation training; and (b) Indic multilingual translation training in various setups (only phase 1 training curves are shown).

data for phase 1 of the training process is the combination of the training datasets for the previous two setups.

In all the four setups described above, the training data for phase 2 is the bilingual data corresponding to the desired language pair. The bilingual data is either the PMI training data or all the available bilingual training data– sizes for which are provided in Table 2.

We now outline the training details for all the setups. We first trained sentencepiece BPE tokenization (Kudo and Richardson, 2018) setting maximum vocabulary size to 32k.[9] The vocabulary was learnt jointly on all the source and target sentence pairs. The number of encoder and decoder layers was set to 3 each, and the number of heads was set to 8. We have considered the hidden size of 128; while the dropout rate was set to 0.1. We initialized the model parameters using Xavier initialization (Glorot and Bengio, 2010). Adam optimizer (Kingma and Ba, 2014) with a learning rate of $5e{-}4$ was used for optimizing model parameters. Gradient clipping was used to clip gradients greater than 1. The training was stopped when the development loss did not improve for 5 consecutive epochs. The same early stopping criterion was followed for both phase 1 and phase 2 of the training process. For phase 1, we used the combination of the development data for all the language pairs in the training data; whereas, for phase 2, we only used the desired language pair's development data. For generating translations, we used greedy decoding where we picked the most likely token at each generation time step. The generation was done token-by-token till the end-of-sentence token is generated or the maximum translation length is reached. The maximum translation length was set to 100.

To compare the training under various setups related to the usage of language tags, we show the perplexity of the training and the development data in Figure 4a. The best (lowest) perplexity is obtained by using the target language tag. However, using the target language tag requires more epochs to converge, where convergence is determined by the early stopping criterion described above.

We show the development BLEU scores, computed using sacreBLEU (Post, 2018) in Table 3 for each language pair. Results indicate that the usage of language tags produces better translation overall. It may also be noted that using both languages' (source and target) tags resulted in the highest development BLEU scores for 8 out of 10 Indic languages while translating to English. For translation from English to Indic languages, the target language tag setup performed the best overall obtaining the highest development BLEU scores in 9 out of 10 languages. We selected the best systems (20 in total) based on the dev BLEU scores for each language pair and used them to generate translations of the test inputs.

The choices related to the hyperparameters that determine the model size and the choice of the training data for phase 1 of the training process were made such that the per epoch

[9]BPE based tokenization performed better in comparison to word-level tokenization using Indic tokenizers (Kunchukuttan, 2020).

Language pair	No tag			Src. tag			Trg. tag			Src. & trg. tags		
	Phase 1	Phase 2		Phase 1	Phase 2		Phase 1	Phase 2		Phase 1	Phase 2	
		PMI	ALL		PMI	ALL		PMI	ALL		PMI	ALL
bn-en	11.8	12.1	11.5	12.9	13.2	11.7	-	-	-	14.1	**14.7**	11.7
gu-en	17.7	17.8	24.4	19.4	19.3	**24.9**	-	-	-	22.7	23.1	23.1
hi-en	18.7	19.6	25.6	21.3	21.6	26.0	-	-	-	25.1	25.7	**26.2**
kn-en	14.5	15.1	16.5	16.6	16.8	15.5	-	-	-	18.7	**19.5**	17.0
ml-en	12.2	12.6	12.2	13.6	13.4	12.3	-	-	-	15.4	**15.9**	12.4
mr-en	13.3	12.9	16.1	14.9	15.1	17.0	-	-	-	16.6	17.2	**17.3**
or-en	14.0	14.1	16.9	15.5	15.6	18.7	-	-	-	17.5	17.8	**20.3**
pa-en	17.4	17.8	**27.0**	18.9	19.0	26.3	-	-	-	22.2	22.8	26.4
ta-en	13.2	13.2	15.0	14.7	14.3	14.6	-	-	-	15.8	**16.4**	15.9
te-en	14.4	14.5	16.5	15.6	16.3	16.8	-	-	-	16.9	**17.9**	16.7
en-bn	-	-	-	-	-	-	6.2	**6.5**	4.6	5.6	5.9	4.4
en-gu	-	-	-	-	-	-	18.4	**19.9**	18.8	16.9	18.4	18.5
en-hi	-	-	-	-	-	-	22.4	24.5	**24.7**	20.6	23.2	24.2
en-kn	-	-	-	-	-	-	12.6	**13.4**	10.6	10.9	12.6	9.8
en-ml	-	-	-	-	-	-	3.9	**4.4**	2.6	3.6	4.0	2.0
en-mr	-	-	-	-	-	-	10.2	**11.2**	10.4	8.8	10.6	10.1
en-or	-	-	-	-	-	-	12.4	13.2	14.0	11.4	12.3	**14.2**
en-pa	-	-	-	-	-	-	18.8	19.7	**20.9**	16.5	18.8	20.5
en-ta	-	-	-	-	-	-	8.5	**9.6**	8.4	7.8	8.3	8.0
en-te	-	-	-	-	-	-	2.2	**2.9**	2.4	2.0	2.6	**2.9**

Table 3: Development BLEU scores for Indic multilingual translations in various setups after phase 1 and phase 2 of the training process. Scores are shown for each language pair separately.

training time is below an hour on a single GPU. We note that there is room for improvement in our results: (a) the model size in any of the setups described earlier can be increased to match the size of the transformer big model (Vaswani et al., 2017), and (b) all the available training data can be used for phase 1 of the training process instead of just the PMI data.

4 Results

System and WAT Task Label	WAT BLEU NLPHut	Best Comp
English→Hindi MM Task		
MMEVTEXT21en-hi	42.11	**44.61**
MMEVHI21en-hi	**1.30**	-
MMCHTEXT21en-hi	43.29	**53.54**
MMCHHI21en-hi	**1.69**	-
English→Malayalam MM Task		
MMEVTEXT21en-ml	**34.83***	30.49
MMEVHI21en-ml	**0.97**	-
MMCHTEXT21en-ml	12.15	**12.98**
MMCHHI21en-ml	**0.99**	-

Table 4: WAT2021 Automatic Evaluation Results for English→Hindi and English→Malayalam. Rows containing "TEXT" in the task label name denote text-only translation track, and the rest of the rows represent image-only track. For each task, we show the score of our system (NLPHut) and the score of the best competitor in the respective task. The scores marked with '∗' indicate the best performance in its track among all competitors.

We report the official automatic evaluation results of our models for all the participated tasks in Table 4 and Table 5. We have provided the automatic evaluation score (BLEU)

WAT Task	From English NLPHut	From English Best Comp	Into English NLPHut	Into English Best Comp
INDIC21en-bn	8.13	**15.97**	13.88	**31.87**
INDIC21en-hi	25.37	**38.65**	24.55	**46.93**
INDIC21en-gu	17.76	**27.80**	23.10	**43.98**
INDIC21en-ml	4.57	**15.49**	15.47	**38.38**
INDIC21en-mr	10.41	**20.42**	17.07	**36.64**
INDIC21en-ta	7.68	**14.43**	15.40	**36.13**
INDIC21en-te	4.88	**16.85**	16.48	**39.80**
INDIC21en-pa	22.60	**33.43**	24.35	**46.39**
INDIC21en-or	12.81	**20.15**	18.92	**37.06**
INDIC21en-kn	11.84	**21.30**	17.72	**40.34**

Table 5: WAT2021 Automatic Evaluation Results for Indic Multilingual Task. For each task, we show the score of our system (NLPHut) and the score of the best competitor ('Best Comp') in the respective task.

for the image captioning task, although it is not apt for evaluating the quality of the generated caption. Thus, we have also provided some sample outputs in Table 6.

5 Conclusions

In this system description paper, we presented our systems for three tasks in WAT 2021 in which we participated: (a) English→Hindi Multimodal task, (b) English→Malayalam Multimodal task, and (c) Indic Multilingual translation task. As the next steps, we plan to explore further on the Indic Multilingual translation task by utilizing all given data and using additional resources for training. We are also working on improving the region-specific image captioning by fine-tuning the object detection model.

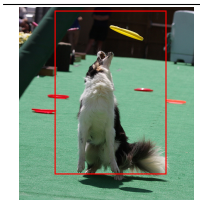
	Gold: एक लड़की टेनिस खेल रही है Gloss: A girl is playing tennis Output:एक टेनिस रैकेट पकड़े हुए आदमी Gloss: A man holding a tennis racket		Gold: आदमी समुद्र में सर्फिंग Gloss: man surfing in ocean Output: पानी में एक व्यक्ति Gloss: A man in the water
	Gold: एक कुत्ता कूदता है Gloss: A dog is jumping Output: कुत्ता भाग रहा है Gloss: A dog is running		Gold: हेलमेट पहनना Gloss: Wearing helmet Output: एक आदमी के सिर पर एक काला हेलमेट Gloss: A black helmet on the head of a person
	Gold: തിളക്കമുള്ള പച്ച കൈറ്റ് Gloss: Bright green kite Output:ആകാശത്ത് പറക്കുന്ന കൈറ്റ് Gloss: Kite flying in the sky		Gold: ഒരു ധ്രുവത്തിലെ ട്രാഫിക് ലൈറ്റ് Gloss: Traffic light at a pole Output: ട്രാഫിക് ലൈറ്റ് ചുവപ്പ് തിളങ്ങുന്നു Gloss: The traffic light glows red
	Gold: തൂങ്ങി കിടക്കുന്ന ഒരു കൂട്ടം വാഴപ്പഴം Gloss: A bunch of hanging bananas Output: ഒരു കൂട്ടം വാഴപ്പഴം Gloss: A bunch of bananas		Gold: ചുമരിൽ ഒരു ഘടികാരം വാഴപ്പഴം Gloss: A clock on the wall Output: ചുമരിൽ ഒരു ചിത്രം Gloss: A picture on the wall

Table 6: Sample captions generated for the evaluation test set using the proposed method: the top two rows present results of Hindi captions; and the bottom two rows are results of Malayalam caption.

Acknowledgments

The authors Shantipriya Parida and Petr Motlicek were supported by the European Union's Horizon 2020 research and innovation program under grant agreement No. 833635 (project ROXANNE: Real-time network, text, and speaker analytics for combating organized crime, 2019-2022). Ondřej Bojar would like to acknowledge the support of the grant 19-26934X (NEUREM3) of the Czech Science Foundation.

The authors do not see any significant ethical or privacy concerns that would prevent the processing of the data used in the study. The datasets do contain personal data, and these are processed in compliance with the GDPR and national law.

References

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Comput. Surv.*, 53(5).

Ross Girshick. 2015. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy. PMLR.

Barry Haddow and Faheem Kirefu. 2020. PMIndia – A Collection of Parallel Corpora of Languages of India. *arXiv e-prints*, page arXiv:2001.09907.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. Cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

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.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language*

Processing: System Demonstrations, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Arun Kumar, Ryan Cotterell, Lluís Padró, and Antoni Oliver. 2017. Morphological analysis of the dravidian language family. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 217–222.

Anoop Kunchukuttan. 2020. The IndicNLP Library. https://github.com/anoopkunchukuttan/indic_nlp_library/blob/master/docs/indicnlp.pdf.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The IIT Bombay English-Hindi Parallel Corpus. *arXiv preprint arXiv:1710.02855*.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *CoRR*, abs/1901.07291.

Annika Lindh, Robert J Ross, Abhijit Mahalunkar, Giancarlo Salton, and John D Kelleher. 2018. Generating diverse and meaningful captions. In *International Conference on Artificial Neural Networks*, pages 176–187. Springer.

Takashi Miyazaki and Nobuyuki Shimizu. 2016. Cross-lingual image caption generation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1780–1790.

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.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Shohei Higashiyama, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, et al. 2020. Overview of the 7th workshop on asian translation. In *Proceedings of the 7th Workshop on Asian Translation*, pages 1–44.

Shantipriya Parida and Ondřej Bojar. 2018. Translating short segments with nmt: A case study in english-to-hindi. In *21st Annual Conference of the European Association for Machine Translation*, page 229.

Shantipriya Parida, Ondřej Bojar, and Satya Ranjan Dash. 2019. Hindi visual genome: A dataset for multi-modal english to hindi machine translation. *Computación y Sistemas*, 23(4).

Shantipriya Parida, Petr Motlicek, Amulya Ratna Dash, Satya Ranjan Dash, Debasish Kumar Mallick, Satya Prakash Biswal, Priyanka Pattnaik, Biranchi Narayan Nayak, and Ondřej Bojar. 2020. Odianlp's participation in wat2020. In *Proceedings of the 7th Workshop on Asian Translation*, pages 103–108.

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

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.

Moses Soh. Learning cnn-lstm architectures for image caption generation.

Raimonda Staniūtė and Dmitrij Šešok. 2019. A systematic literature review on image captioning. *Applied Sciences*, 9(10):2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Qi Wu, Chunhua Shen, Peng Wang, Anthony Dick, and Anton Van Den Hengel. 2017. Image captioning and visual question answering based on attributes and external knowledge. *IEEE transactions on pattern analysis and machine intelligence*, 40(6):1367–1381.

Zhishen Yang and Naoaki Okazaki. 2020. Image caption generation for news articles. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1941–1951.

Zhongliang Yang, Yu-Jin Zhang, Sadaqat ur Rehman, and Yongfeng Huang. 2017. Image captioning with object detection and localization. In *International Conference on Image and Graphics*, pages 109–118. Springer.

Jun Yu, Jing Li, Zhou Yu, and Qingming Huang. 2019. Multimodal transformer with multi-view visual representation for image captioning. *IEEE transactions on circuits and systems for video technology*, 30(12):4467–4480.