NICT's Participation in WAT 2018: Approaches Using Multilingualism and Recurrently Stacked Layers

Raj Dabre
NICT,
3-5 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0289, Japan
raj.dabre@nict.go.jp

Anoop Kunchukuttan
Microsoft AI and Research, India
ankunchu@microsoft.com

Atsushi Fujita
NICT,
3-5 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0289, Japan
atsushi.fujita@nict.go.jp

Eiichiro Sumita
NICT,
3-5 Hikaridai, Seika-cho, Soraku-gun,
Kyoto 619-0289, Japan
eiichiro.sumita@nict.go.jp

Abstract

In this paper we describe all our NMT systems for the following translation tasks we participated in: ASPEC (all tasks), Indic Languages (multilingual tasks) and the Myanmar-English task. Our team,"NICT-5", focused on the utility of bidirectional, recurrently stacked layered and multilingual models for ASPEC, simple domain adaptation approaches for Myanmar-English and multilingual models for Indic Languages. In the case of ASPEC translation, we noted that a single multilingual/bidirectional model (without ensembling) has the potential to achieve (near) state-of-the-art results for all the language pairs. We also noted that models that use recurrently stacking layers do not experience a large loss in translation quality despite having significantly fewer parameters compared to the vanilla NMT models. An interesting observation is that systems with the best BLEU might not be the best in terms of human evaluation.

1 Introduction

Neural machine translation (NMT) (Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015) has enabled end-to-end training of a translation system without needing to deal with word alignments, translation rules, and complicated decoding algorithms, which are the characteristics of phrase-based statistical machine translation (PBSMT) (Koehn et al., 2007). Although vanilla NMT is significantly better than PBSMT in resource-rich scenarios, PBSMT performs better in resource-poor scenarios (Zoph et al., 2016). By exploiting transfer learning techniques, the performance of NMT approaches can be improved substantially.

For WAT 2018, we participated as team "NICT-5" and worked on ASPEC Chinese-Japanese and English-Japanese translation, UCSY Myanmar-English translation and Indic multilingual translation directions. The techniques we focused on for each translation task can be summarized as below:

- For the ASPEC translation tasks, we mostly relied on multilingual Transformer (Vaswani et al., 2017) models and experimented with Recurrently Stacked NMT (RS-NMT) (Dabre and Fujita, 2018) models in order to determine the trade-off between compactness of models and the loss in their performance.

- For the UCSY Myanmar-English translation task, we tried domain adaptation techniques such as Mixed Fine Tuning (Chu et al., 2017) since the the final objective was to achieve high quality translation for a low-resource domain (ALT).

- For the Indic multilingual task, we explored the feasibility of bilingual, N-to-1, 1-to-N and N-to-N way translation models. We also tried an approach where we mapped the scripts of all Indic languages to a common script (Devanagari) to see if it helps improve the performance of a multilingual model.

For additional details of how our submissions are ranked relative to the submissions of other WAT

32nd Pacific Asia Conference on Language, Information and Computation
The 5th Workshop on Asian Translation
Hong Kong, 1-3 December 2018

participants, kindly refer to the overview paper (Nakazawa et al., 2018).

2 NMT Models and Approaches

We will first describe the Transformer which is the state-of-the-art NMT model we used for our experiments.

2.1 The Transformer

The Transformer (Vaswani et al., 2017) is the current state-of-the-art model for NMT. It is a sequence-to-sequence neural model that consists of two components, the *encoder* and the *decoder*. The encoder converts the input word sequence into a sequence of vectors of high dimensionality. The decoder, on the other hand, produces the target word sequence by predicting the words using a combination of the previously predicted word and relevant parts of the input sequence representations. Due to lack of space, we briefly describe the encoder and decoder as follows. The reader is encouraged to read the Transformer paper (Vaswani et al., 2017) for a deeper understanding.

Suppose that X and Y are the input and output word sequences where $X = [x_0, x_1, ..., x_n]$ and $Y = [y_0, y_1, ..., y_m]$. The objective is to predict the best word sequence:

$$\hat{Y} = \arg\max_{Y} P(Y|X).$$

The first step is to compute initial high dimensional vector space representations E_X for the input word sequence X by using its word embeddings where:

$$E_X^{encoder} = embedding_{encoder}(X).$$

Positional information is also incorporated into the word embeddings, which has been shown to be important for good performance. The embeddings, computed by an embedding layer, are stored in an embedding matrix in which the number of rows is equal to the size of the vocabulary of the input sequence. These word embeddings are then processed by N neural network layers composed of self attention, feed-forward layers and normalization sublayers which help produce a high-level representation of the input sequence denoted by S_X^N where:

$$S_X^i = Layer_{encoder}(S_X^{i-1}),$$

and $S_X^0 = E_X^{encoder}$. These processing steps describe the Encoder.

Due to the use of self-attention layer instead of the traditional recurrent layer, the input sequence can be processed in parallel, the result of which is significantly faster processing because of parallel computation.

The decoder, on the other hand, predicts one word at a time by taking into account the previously predicted words. At each time step i, the decoder takes the previously predicted word y_{i-1}, computes its embedding as:

$$E_{y_{i-1}}^{decoder} = embedding_{decoder}(y_{i-1}),$$

processes this embedding through N layers of self-attention, cross-attention (to access the relevant information from the source hidden-state representations) to give the decoder hidden-state s_i. s_i is then converted into a probability distribution to predict y_i:

$$y_i = argmax_{y_i} P(y_i|X, y_{i-1}, y_{i-2}, ..., y_0).$$

The distribution is obtained using a softmax layer as follows:

$$P(y_i|X, y_{i-1}, y_{i-2}, ..., y_0) = softmax(W^T \times s_i),$$

where W is a matrix which maps s_i to a vector of the size of the vocabulary of the target sequence. $W^T \times s_i$ is also known as the logit vector for the i^{th} word to be predicted, denoted as L_i.

2.2 Multilingualism in NMT

In order to train multilingual models using the Transformer, we used the artificial token based approach (Johnson et al., 2017) which is also useful for zero-shot translation. We simply concatenate the corpora for all translation directions after inserting a token like “XX” at the beginning of the source sentence, where “XX” is a special token that indicates the target language such as JA (for Japanese) or EN (for English). “XX” should be a token which is not already present in the corpus. We also over-sample the smaller corpora to match the size of the larger corpora.

2.3 Domain Adaptation in NMT

The primary domain adaptation technique we explored was Mixed Fine Tuning (MFT) (Chu et al., 2017) since it can be used without any modification to the model architecture. The approach can be summarized as follows:

- Learn a joint vocabulary for the out-of-domain and in-domain corpora.
- Train the NMT model on the out-of-domain corpus only until convergence.
- Resume training the same NMT model on the combination of out-of-domain and in-domain corpora[1] till convergence.

This method is extremely easy to use.

2.4 Recurrently Stacked Layers in NMT

Recurrently Stacked Layers for NMT (RS-NMT) (Dabre and Fujita, 2018) proposes to reuse share the parameters of among all layers of the encoder or the decoder. A N-layer (encoder-decoder) RS-NMT has the same number of parameters as a 1-layer vanilla NMT model. The only major difference is that the same layer is recurrently stacked N times for each of the encoder and the decoder. As a result of recurrently stacking layers, the NMT model learns to refine the representations of sentences leading to significantly better performance than a vanilla 1-layer model. A generalized version of this approach is proposed in the work on the Universal Transformer (Dehghani et al., 2018). In this work a 6-layer transformer is recurrently stacked M times where M is dynamically decided. The major difference between RS-NMT and Universal Transformers is that the former seeks to reduce the number of model parameters with minimal loss in performance whereas the latter seeks to improve the model performance over its vanilla counterpart without increasing the number of parameters.

In this paper, we choose an intermediate approach where we consider a 3-layer transformer and recurrently stack it 4 times. We do not opt for the dynamic recurrent stacking approach for simplicity.

Split	ASPEC JC	ASPEC JE
Train	672,315	3,008,500
Dev	2,090	1,790
Test	2,107	1,812

Table 1: ASPEC dataset splits. The number indicates the number of lines in the split.

3 Model Training Details

Since we pre-processed all our data according to the organizer's guidelines, we do not mention them here. For all our experiments, we used the tensor2tensor[2] version 1.6 implementation of the Transformer (Vaswani et al., 2017) model. We chose this implementation because it is known to give the state-of-the-art results for NMT. In order to train multilingual models we used the artificial token trick used for zero-shot NMT (Johnson et al., 2017). We always oversample the smaller datasets to ensure that the training phase sees equal amounts of data from all datasets. We also modified the tensor2tensor implementation to train Recurrently Stacked NMT (RS-NMT) models (Dabre and Fujita, 2018). We used the default hyperparameters in tensor2tensor for all our models with the exception of the number of training iterations. Unless mentioned otherwise we use the "base" transformer model hyperparameter settings with a 32000 subword vocabulary which is learned using tensor2tensor's default subword segmentation mechanism. During training, a model checkpoint is saved every 1000 iterations. We averaged the last 10 model checkpoints and used it for decoding the test sets.

4 ASPEC Task

4.1 Datasets

For the ASPEC (Nakazawa et al., 2016) tasks we used the official data provided by the organizers. The objective of the ASPEC task is to push the state-of-the-art for scientific domain machine translation for Japanese-English and Japanese-Chinese. The parallel corpora available, belong to the scientific domain and are sufficiently large in size. Refer to Table 1 for an overview of the data splits. For our ex-

[1] The in-domain data will be oversampled so that the training phase sees equal amounts of data from both domains.

[2] https://github.com/tensorflow/tensor2tensor

Task	Model	Our BLEU	Top BLEU	BLEU Ranking	Human Ranking
English-Japanese	Bidirectional	42.87	43.43	2/6	1/5
English-Japanese	RS-NMT*	41.91	43.43	-	-
Japanese-English	Multilingual	29.65	30.59	2/4	1/4
Japanese-English	Unidirectional*	28.63	30.59	-	-
Chinese-Japanese	Multilingual	49.79	49.79	1/2	1/1
Chinese-Japanese	MFT*	49.67	49.79	-	-
Japanese-Chinese	Multilingual	35.99	37.60	2/3	2/2
Japanese-Chinese	Vanilla*	35.71	37.60	-	-

Table 2: ASPEC task results. Entries with an asterisk mark are the comparative submissions and hence are not considered in the overall ranking.

Split	UCSY	ALT
Train	208,638	17,965
Dev	-	993
Test	-	1,007

Table 3: Myanmar-English dataset splits. The number indicates the number of lines in the split.

periments we used all the data for ASPEC Japanese-Chinese but for Japanese-English we used only the top 1.5 million lines since the bottom half of the corpus is of poorer quality and contains many badly aligned segments.

4.2 Models Trained

For the ASPEC task we trained vanilla and recurrently stacked NMT models for unidirectional translation. We used shared encoder-decoder vocabularies for multilingual (including bidirectional) models and for Chinese-Japanese models in order to enable cognate sharing. Kindly note that in this paper, unidirectional models can only translate in one direction and bidirectional models can translate in both directions. Our usage of the words uni and bidirectional are not related to the bidirectional RNNs used in the traditional seq2seq models. For vanilla English-Japanese models we use separate vocabularies because there is no scope for cognate sharing. For each translation direction we submitted two types of models which are as follows:

- English to Japanese: a. A bidirectional transformer using the top 1.5M lines of the English-Japanese corpus. This model uses the "big" model hyperparameter setting as defined in the original paper. b. A RS-NMT model in which a 3 layer transformer was recurrently stacked 4 times.[3] This model was trained using the whole corpus of 3M lines. Both models were trained for 300k iterations.

- Japanese to English: a. A multilingual transformer that uses 1.5M lines of English-Japanese and the whole Japanese-Chinese corpus. This single transformer model can translate both two and from English and Japanese as well as Japanese and Chinese. This transformer also uses the "big" model hyperparameter settings. b. A unidirectional Japanese to English transformer using the full 3M lines corpus.

- Chinese to Japanese: a. The same multilingual model used for Japanese to English translation. b. A model that uses mixed fine tuning by first training on 3M lines of En-Ja for 200k iterations followed by an additional 100k iterations on a combined dataset of 3M lines of En-Ja.

- Japanese to Chinese: a. The same multilingual model used for Japanese to English translation. b. The vanilla transformer model.

4.3 Results

Refer to Table 2 for an overview of the ASPEC task results. In general our submissions secured second rank in terms of BLEU for 3/4 tasks and first rank

[3] In the original RS-NMT model (Dabre and Fujita, 2018) a single layer is recurrently stacked N times. As such the number of parameters is the same as a 1-layer transformer. However in our case the number of parameters is the same as a 3-layer transformer.

Split	Bengali	Hindi	Malayalam	Tamil	Telugu	Urdu	Sinhalese
Train	337,428	337,428	359,423	26,217	22,165	26,619	521,726
Dev	500	500	500	500	500	500	500
Test	1,000	1,000	1,000	1,000	1,000	1,000	1,000

Table 4: Indic languages dataset splits. The number indicates the number of lines in the split.

in terms of human evaluation for 3/4 tasks. It is important to note that the best performing submission (in terms of BLEU) for 3/4 tasks used ensembling (possibly on top of checkpoint averaging) for decoding whereas our submissions used only checkpoint averaging. In the case of Japanese-Chinese, the submission with the best BLEU used an ensemble of 10 models. In comparison our submissions involve only one model and hence we believe that our models are better suited for deployment in a practical scenario. It is also important to note that our English-Japanese RS-NMT model is within 1 BLEU point of our best submission. This shows that recurrent stacking of layers is quite formidable and deserves plenty of exploration in the future. For reference, the number of parameters in the 6-layer English-Japanese model is 283,313,671 (BLEU 41.31) whereas for the RS-NMT model this number drops to 217,171,975 (BLEU 41.31). It is clear that RS-NMT can help create compact models without loss in translation quality.

5 Myanmar to English Translation Task

5.1 Datasets

The Myanmar-English datasets consist of parallel corpora from two different domains. The objective of the Myanmar-English translation task is to improve the translation quality for the ALT (Asian Language Treebank) (Riza et al., 2016) which consists of relatively low-resource language pairs. For this domain the parallel corpus is extremely small. As such, a larger out-of-domain corpus for the same language pair also known as the UCSY[4] corpus is provided. Refer to Table 3 for the corpora splits.

5.2 Models Trained

For Myanmar to English translation, we submitted only one model which was trained using mixed fine tuning (MFT). We first trained a model on the out-of-domain UCSY data for 100000 iterations followed by training for a further 20000 iterations on a combination of UCSY and ALT data. Since this is a low-resource setting we used separate encoder-decoder vocabularies of 16k subwords.

5.3 Results

For Myanmar-English the best BLEU score we obtained was 15.44 using only UCSY and ALT data by training the model using mixed fine tuning (MFT). In comparison the best performing submission had a BLEU of 29.14. It should be noted that this system used additional monolingual data. Our BLEU based ranking is 3/6 and our human evaluation based ranking is 3/5. In the future we will explore more sophisticated mechanisms for this language pair and adopt the use of monolingual corpora which is shown to be highly effective.

6 Indic Languages Task

6.1 Datasets

The Indic language dataset spans 8 languages, 7 of which are Indic languages and one of them being English. The objective of the Indic shared task is to test the feasibility of multilingualism for low-resource machine translation for related languages. The Indic languages involved are Bengali, Hindi, Malayalam, Tamil, Telugu, Sinhalese and Urdu. The corpus belongs to the OpenSubtitles domain. Refer to Table 4 for the corpora splits. Although monolingual corpora were provided, we did not use them.

6.2 Models Trained

We trained the following five types of models:

- Unidirectional models: We trained separate vocabulary models for translating from English to the Indic languages and from the Indic languages to English. The vocabulary size was 8k and separate encoder-decoder vocabularies

[4] http://www.nlpresearch-ucsy.edu.mm/

Task	Model	Our BLEU	Top BLEU	BLEU Ranking	Human Ranking
English-Bengali	uni	14.55	18.81	-	-
English-Bengali	En-XX	10.45	18.81	-	-
English-Bengali	XX-YY	10.39	18.81	-	-
English-Hindi	uni	26.35	44.08	-	-
English-Hindi	En-XX	29.65	44.08	2/4	-
English-Hindi	XX-YY	26.59	44.08	-	1/4
English-Malayalam	uni	16.56	16.56	-	-
English-Malayalam	En-XX	7.29	16.56	-	-
English-Malayalam	XX-YY	4.87	16.56	-	-
English-Tamil	uni	8.74	30.53	-	-
English-Tamil	En-XX	18.60	30.53	-	-
English-Tamil	XX-YY	20.39	30.53	2/4	3/3
English-Telugu	uni	10.74	41.89	-	-
English-Telugu	En-XX	25.64	41.89	-	-
English-Telugu	XX-YY	29.17	41.89	-	-
English-Urdu	uni	20.21	32.86	-	-
English-Urdu	En-XX	27.05	32.86	-	-
English-Urdu	XX-YY	29.05	32.86	-	-
English-Sinhalese	uni	9.78	18.09	-	-
English-Sinhalese	En-XX	8.35	18.09	-	-
English-Sinhalese	XX-YY	7.51	18.09	-	-

Table 5: Indic task results for translation from English to Indic Languages. We only consider the ranking of our submissions with the highest BLEU/human scores. As such, some submissions have a BLEU ranking but not a human ranking and vice versa. Note that, only English-Hindi, Hindi-English, English-Tamil and Tamil-English translation directions were submitted for human evaluation.

were used. Due to lack of time required for hyperparameter tuning, we trained the models for 100k iterations on the default model setting.

- Multilingual XX-En model: We trained a single model to translate from all the Indic languages to English by combining all the training data. Since the target language is the same, this is the only multilingual model that does not need artificial tokens to indicate the target language. We trained this model for 500k iterations.

- Multilingual En-XX model: We trained a single model to translate from English to all the Indic languages. This is essentially the reverse of the XX-En model. We also trained this model for 500k iterations.

- Multilingual XX-YY model: We trained a single model to translate from all the Indic languages to English and vice versa. Unlike the previous multilingual models, we trained this model only for 180k iterations due to lack of time.

- Multilingual Shared Indic Script XX-En model: This model is similar to the XX-En model except that the scripts for all the Indic languages are mapped to a common script. We used Devanagari as the common script, and used the *Indic NLP Library*[5] (Kunchukuttan et al., 2015) for script conversion. As such, this increases the chance of vocabulary sharing. Because the training corpus diversity is significantly reduced we trained this model for 100k iterations because it is technically equivalent

[5] https://github.com/anoopkunchukuttan/indic_nlp_library/

Task	Model	Our BLEU	Top BLEU	BLEU Ranking	Human Ranking
Bengali-English	uni	19.17	20.05	-	-
Bengali-English	XX-En	18.03	20.05	-	-
Bengali-English	UXX-En	18.82	20.05	-	-
Bengali-English	XX-YY	16.68	20.05	-	-
Hindi-English	uni	26.05	32.95	2/4	-
Hindi-English	XX-En	31.06	32.95	-	-
Hindi-English	UXX-En	31.51	32.95	-	-
Hindi-English	XX-YY	30.21	32.95	-	2/4
Malayalam-English	uni	22.87	22.87	-	-
Malayalam-English	XX-En	14.06	22.87	-	-
Malayalam-English	UXX-En	15.91	22.87	-	-
Malayalam-English	XX-YY	10.90	22.87	-	-
Tamil-English	uni	11.09	24.31	-	-
Tamil-English	XX-En	21.37	24.31	-	-
Tamil-English	UXX-En	21.27	24.31	-	-
Tamil-English	XX-YY	24.31	24.31	1/4	2/4
Telugu-English	uni	15.76	33.23	-	-
Telugu-English	XX-En	29.85	33.23	-	-
Telugu-English	UXX-En	30.23	33.23	-	-
Telugu-English	XX-YY	33.23	33.23	-	-
Urdu-English	uni	20.65	30.84	-	-
Urdu-English	XX-En	27.88	30.84	-	-
Urdu-English	UXX-En	26.73	30.84	-	-
Urdu-English	XX-YY	30.84	30.84	-	-
Sinhalese-English	uni	21.85	21.85	-	-
Sinhalese-English	XX-En	18.73	21.85	-	-
Sinhalese-English	UXX-En	19.19	21.85	-	-
Sinhalese-English	XX-YY	17.25	21.85	-	-

Table 6: Indic task results for translation from Indic Languages to English. We only consider the ranking of our submissions with the highest BLEU/human scores. As such, some submissions have a BLEU ranking but not a human ranking and vice versa. Note that, only English-Hindi, Hindi-English, English-Tamil and Tamil-English translation directions were submitted for human evaluation.

to a unidirectional translation model.

6.3 Results

Refer to Tables 5 and 6 for the results for the Indic Languages translation task. We give the results for the unidirectional (uni), multitarget (En-XX), multisource (XX-En), multilingual shared Indic script (UXX-En) and multisource multitarget (XX-YY) models. In case of translation to the Indic languages, our best submissions was able to secure 2nd rank (BLEU) for most language pairs. On the other hand, for translation from the Indic languages, our best submissions were able to secure 1st rank (BLEU) for most language pairs. In terms of human evaluation we also managed to secure 2nd rank most of the times for the language pairs that were submitted.

From our results it is clear that multilingual models do not work perfectly well for all translation directions but they do help in improving translation quality for the low-resource languages (Hindi, Tamil, Urdu and Telugu). Multilingual models led to poor performance for the resource rich translation directions. As expected, vocabulary unification did lead to slight improvements in translation quality and we believe that such approaches deserve further exploration. Due to lack of time we did not try advanced multilingual models and training approaches which we expect will lead to a single multilingual model that will perform well for all translation directions.

7 Conclusion

In this paper we have described our submissions to WAT 2018. We managed to obtain near state-of-the-art results for ASPEC and showed the effectiveness of multilingual transformers. We also showed that recurrently stacked NMT models can lead to high quality models while significantly reducing the number of model parameters. We explored the utility of mixed fine tuning for Myanmar-English translation. We also studied the impact of multilingualism in the case of Indic languages translation.

Overall we have observed that multilingualism leads to significant improvements in translation quality and reduce the need to train multiple translation models effectively leading to parameter reduction. We have also observed that BLEU score based ranking is mostly inconsistent with human evaluation based ranking, especially for the ASPEC task. We believe that this calls for innovation into newer automatic evaluation metrics that correlate well with human evaluations. In the future we plan to explore more sophisticated multilingual models which should help push the state-of-the-art even further.

Acknowledgments

We would like to thank the reviewers for their comments which helped improve the quality of this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, USA, May. International Conference on Learning Representations.

Kyunghyun Cho, Bart van Merriënboer, Çalar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, Vancouver, Canada, July. Association for Computational Linguistics.

Raj Dabre and Atsushi Fujita. 2018. Recurrent stacking of layers for compact neural machine translation models. *CoRR*, abs/1807.05353.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. 2018. Universal transformers. *CoRR*, abs/1807.03819.

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June. Association for Computational Linguistics.

Anoop Kunchukuttan, Ratish Puduppully, and Pushpak Bhattacharyya. 2015. Brahmi-Net: A transliteration and script conversion system for languages of the Indian subcontinent. In *Conference of the North American Chapter of the Association for Computational Linguistics - Human Language Technologies: System Demonstrations*.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*, Paris, France, May. European Language Resources Association (ELRA).

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Raj Dabre, Anoop Kunchukuttan, Win Pa Pa, Isao Goto, Hideya Mino, Katsuhito Sudoh, and Sadao Kurohashi. 2018. Overview of the 5th workshop on asian translation. In *Proceedings of the 5th Workshop* *on Asian Translation (WAT2018)*, Hong Kong, China, December.

H. Riza, M. Purwoadi, Gunarso, T. Uliniansyah, A. A. Ti, S. M. Aljunied, L. C. Mai, V. T. Thang, N. P. Thai, V. Chea, R. Sun, S. Sam, S. Seng, K. M. Soe, K. T. Nwet, M. Utiyama, and C. Ding. 2016. Introduction of the asian language treebank. In *2016 Conference of The Oriental Chapter of International Committee for Coordination and Standardization of Speech Databases and Assessment Techniques (O-COCOSDA)*, pages 1–6, Oct.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems*, NIPS'14, pages 3104–3112, Cambridge, MA, USA. MIT Press.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1568–1575.