Language Relatedness and Lexical Closeness can help Improve Multilingual NMT: IITBombay@MultiIndicNMT WAT2021

Jyotsana Khatri, Nikhil Saini, Pushpak Bhattacharyya
Department of Computer Science and Engineering
Indian Institute of Technology Bombay
Mumbai, India
{jyotsanak, nikhilra, pb}@cse.iitb.ac.in

Abstract

Multilingual Neural Machine Translation has achieved remarkable performance by training a single translation model for multiple languages. This paper describes our submission (Team ID: CFILT-IITB) for the MultiIndicMT: An Indic Language Multilingual Task at WAT 2021. We train multilingual NMT systems by sharing encoder and decoder parameters with language embedding associated with each token in both encoder and decoder. Furthermore, we demonstrate the use of transliteration (script conversion) for Indic languages in reducing the lexical gap for training a multilingual NMT system. Further, we show improvement in performance by training a multilingual NMT system using languages of the same family, i.e., related languages.

1 Introduction

Neural Machine Translation (Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016) has become a de-facto for automatic translation of language pairs. NMT systems with Transformer (Vaswani et al., 2017) based architectures have achieved competitive accuracy on data-rich language pairs like English-French. However, NMT systems are data-hungry, and only a few pairs of languages have abundant parallel data. For low resource setting, techniques like transfer learning (Zoph et al., 2016) and utilization of monolingual data in an unsupervised setting (Artetxe et al., 2018; Lample et al., 2017, 2018) have shown support for increasing the translation accuracy. Multilingual Neural Machine Translation is an ideal setting for low resource MT (Lakew et al., 2018) since it allows sharing of encoder-decoder parameters, word embeddings, and joint or separate vocabularies. It also enables zero-shot translations, i.e., translating between language pairs that were not seen during training (Johnson et al., 2017a).

In this paper, we present our system for MultiIndicMT: An Indic Language Multilingual Task at WAT 2021 (Nakazawa et al., 2021). The task covers 10 Indic Languages (Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, and Telugu) and English.

To summarize our approach and contributions, we **(i)** present a multilingual NMT system with shared encoder-decoder framework, **(ii)** show results on many-to-one translation, **(iii)** use transliteration to a common script to handle the lexical gap between languages, **(iv)** show how grouping of languages in regard to their language family helps multilingual NMT and **(v)** use language embeddings with each token in both encoder and decoder.

2 Related work

2.1 Neural Machine Translation

Neural Machine Translation architectures consist of encoder layers, attention layers, and decoder layers. NMT framework takes a sequence of words as an input; the encoder generates an intermediate representation, conditioned on which, the decoder generates an output sequence. The decoder also attends to the encoder states. Bahdanau et al. (2015) introduced the encoder-decoder attention to allow the decoder to soft-search the parts of the source sentence to predict the next token. The encoder-decoder can be a LSTM framework (Sutskever et al., 2014; Wu et al., 2016), CNN (Gehring et al., 2017), or Transformer layers (Vaswani et al., 2017). A Transformer layer comprises of self-attention that bakes the understanding of input sequence with positional encoding and passes on to the next component, feed-forward neural network, layer normalization, and residual connections. The decoder in the transformer has an additional encoder-attention layer that attends to the output states of the transformer encoder.

Proceedings of the 8th Workshop on Asian Translation, pages 217–223

NMT is data-hungry, and only a few pairs of languages have abundant parallel data. In recent years, NMT has been accompanied by several techniques to improve the performance of both low & high resource language pairs. Back-translation (Sennrich et al., 2016b) is used to augment the parallel data with synthetically generated parallel data by passing monolingual datasets to the previously trained models. Currently, NMT systems also perform on-the-fly back-translation to train the model simultaneously. Tokenization methods like Byte Pair Encoding (Sennrich et al., 2016a) are used in almost all NMT models. Pivoting (Cheng et al., 2017) and Transfer Learning (Zoph et al., 2016) have leveraged the language relatedness by indirectly providing the model with more parallel data from related language pairs.

2.2 Multilingual Neural Machine Translation

Multilingual NMT trains a single model utilizing data from multiple language-pairs to improve the performance. There are different approaches to incorporate multiple language pairs in a single system, like multi-way NMT, pivot-based NMT, transfer learning, multi-source NMT and, multilingual NMT (Dabre et al., 2020). Multilingual NMT came into picture because many languages share certain amount of vocabulary and share some structural similarity. These languages together can be utilized to improve the performance of NMT systems. In this paper, our focus is to analyze the performance of multi-source NMT. The simplest approach is to share the parameters of NMT model across multiple language pairs. These kinds of systems work better if languages are related to each other. In Johnson et al. (2017b), the encoder, decoder, and attention are shared for the training of multiple language pairs and a target language token is added at the beginning of target sentence while decoding. Firat et al. (2016) utilizes a shared attention mechanism to train multilingual models. Recently many approaches have been proposed, where monolingual data of multiple languages is utilized to pre-train a single model using different objectives like masked language modeling and denoising (Lample and Conneau, 2019; Song et al., 2019; Lewis et al., 2020; Liu et al., 2020). Multilingual pre-training followed by multilingual fine-tuning has also proven to be beneficial (Tang et al., 2020).

2.3 Language Relatedness

Telugu, Tamil, Kannada, and Malayalam are Dravidian languages whose speakers are predominantly found in South India, with some speakers in Sri Lanka and a few pockets of speakers in North India. The speakers of these languages constitute around 20% of the Indian population (Kunchukuttan and Bhattacharyya, 2020). Dravidian languages are agglutinative, i.e., long and complex words are formed by stringing together morphemes without changing them in spelling or phonetics. Most Dravidian languages have clusivity distinction. Hindi, Bengali, Marathi, Gujarati, Oriya, Punjabi are Indo-Aryan languages and are primarily spoken in North and Central India and the neighboring countries of Pakistan, Nepal, and Bangladesh. The speakers of these languages constitute around 75% of the Indian population. Both Dravidian and Indo-Aryan language families follow the Subject(S)-Object(O)-Verb(V) order.

Grouping languages concerning their families have inherent advantages because they form a closely related group with several linguistic phenomenons shared amongst them. Indo-Aryan languages are morphologically rich and have huge similarities when compared to English. A language group also share vocabularies at both word and character level. They contain similarly spelled words that are derived from the same root. ‘

2.4 Transliteration

Indic languages share a lot of vocabulary, but most languages utilize different scripts. Nevertheless, these scripts have phoneme overlap and can be converted easily from one to another using a simple rule-based system. To convert all Indic language data into the same script, we use IndicNLP[1] which maps different Unicode range for the conversion. The conversion of all Indic language scripts to the same script helps with better shared vocabulary and leads to smaller subword vocabulary (Ramesh et al., 2021).

3 System overview

In this section, we describe the details of the submitted systems to MultiIndicMT task at WAT2021. We report results for four types of models:

- **Bilingual**: Trained only using parallel data for a particular language pair (bilingual models).

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

- **All-En**: Multilingual many-to-one system trained using all available parallel data of all language pairs.
- **IA-En**: Multilingual many-to-one system trained using Indo-Aryan languages from the provided parallel data.
- **DR-En**: Multilingual many-to-one system trained using Dravidian languages from the provided parallel data.

To train our multilingual models, we use shared encoder-decoder transformer architecture. To handle the lexical gap between Indic languages in multilingual models, we convert the data of all Indic languages to a common script. We choose the common script as Devanagari (arbitrary choice). We also perform a comparative study of systems when the encoder and decoder are shared only between related languages. To perform this comparative study, we group the provided set of languages in two parts based on the language families they belong to, i.e, the system is trained from Indo-Aryan (group) to English, and Dravidian (group) to English. Indo-Aryan-to-English contains Bengali, Gujarati, Hindi, Marathi, Oriya, Punjabi to English, and Dravidian-to-English contains Kannada, Malayalam, Tamil, Telugu to English. We use shared subword vocabulary of the languages involved while training multilingual models, and a common vocabulary of source and target languages to train bilingual models.

4 Experimental details

4.1 Dataset

Our models are trained using only the parallel data provided for the task. The size of the parallel data available and its source of origin are summarized in Table 1. The validation and test data provided in the task is n-way and contains 1000 sentences for validation and 2390 sentences in test set.

4.2 Data preprocessing

We tokenize English language data using moses tokenizer (Koehn et al., 2007), and Indian language data using IndicNLP[2] library. For multilingual models, we transliterate (script mapping) all Indic language data into Devanagari script using the IndicNLP library. Our aim here is to convert data of all languages into the same script, hence the choice of Devnagari as a common script is arbitrary. We use fastBPE[3] to learn BPE (Byte pair encoding) (Bojanowski et al., 2017). For bilingual models, we use 60000 BPE codes over the combined tokenized data of both languages. The number of BPE codes is set to 100000 for All-En, and 80000 for DR-En and IA-En.

4.3 Experimental Setup

We use six layers in the encoder, six layers in the decoder, 8 attention heads in both encoder and decoder, and 1024 embedding dimension. The encoder and decoder are trained using Adam (Kingma and Ba, 2015) optimizer with inverse square root learning rate schedule. We use the same setting as used in Song et al. (2019) for warmup phase, in which the learning rate is increased linearly for some initial steps starting from $1e^{-}7$ to 0.0001, warmup phase is set to 4000 steps. We use mini-batches of size 2000 tokens and set the dropout to 0.1 (Gal and Ghahramani, 2016). Maximum sentence length is set to 100 after applying BPE. At decoding time, we use greedy decoding. For experiments, we are using mt_steps from MASS[4] codebase. Our models are trained using only parallel data provided in the task, we are not training the model using any kind of pretraining objective. We train bilingual models for 100 epochs and multilingual models for 150 epochs. The epoch size is set to 200000 sentences. Due to resource constraints, we train our model for fixed number of epochs, it does not guarantee convergence. Similar to MASS (Song et al., 2019), language embeddings are added to each token in the encoder and decoder to distinguish between languages. These language embeddings are learnt during training.

4.4 Results and Discussion

We report BLEU scores for our four settings: bilingual, All-En (multilingual many-to-one), IA-En (multilingual many-to-one Indo-Aryan to English), and DR-En (multilingual many-to-one Dravidian to English) in Table 2. We use multi-bleu.perl [5] to calculate BLEU scores of baseline models. BLEU score is calculated using the tokenized reference and hypothesis files as followed by organizers in

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

[3] https://github.com/glample/fastBPE

[4] https://github.com/microsoft/MASS

[5] https://github.com/moses-smt/mosesdecoder/blob/RELEASE-2.1.1/scripts/generic/multi-bleu.perl

Lang Pair	Size	Data sources
bn-en	1.70M	alt, cvit-pib, jw, opensubtitles, pmi, tanzil, ted2020, wikimatrix
gu-en	0.51M	bibleuedin, cvit, jw, pmi, ted2020, urst, wikititles
hi-en	3.50M	alt, bibleuedin, cvit-pib, iitb, jw, opensubtitles, pmi, tanzil, ted2020, wikimatrix
kn-en	0.39M	bibleuedin, jw, pmi, ted2020
ml-en	1.20M	bibleudein, cvit-pib, jw, opensubtitles, pmi, tanzil, ted2020, wikimatrix
mr-en	0.78M	bibleuedin, cvit-pib, jw, pmi, ted2020, wikimatrix
or-en	0.25M	cvit, mtenglish2odia, odiencorp, pmi
pa-en	0.51M	cvit-pib, jw, pmi, ted2020
ta-en	1.40M	cvit-pib, jw, nlpc, opensubtitles, pmi, tanzil, ted2020, ufal, wikimatrix, wikititles
te-en	0.68M	cvit-pib, jw, opensubtitles, pmi, ted2020, wikimatrix

Table 1: Parallel Dataset amongst 10 Indic-English language pairs. *Size* is the number of parallel sentences (in millions). (bn, gu, hi, kn, ml, mr, or, pa, ta, te and en: Bengali, Gujarati, Hindi, Kannada, Malayalam, Marathi, Oriya, Punjabi, Tamil, Telugu and English respectively

	BLEU				AMFM		
Lang Pair	**Bilingual**	**IA-En**	**DR-En**	**All-En**	**IA-En**	**DR-En**	**All-En**
bn-en	18.52	**20.18**	-	18.48	**0.734491**	-	0.730379
gu-en	26.51	**31.02**	-	28.79	**0.776935**	-	0.765441
hi-en	33.53	**33.7**	-	30.9	**0.791408**	-	0.775032
mr-en	21.28	**25.5**	-	23.57	**0.767347**	-	0.751917
or-en	22.6	**26.34**	-	25.05	**0.780009**	-	0.770941
pa-en	29.92	**32.34**	-	29.87	**0.782112**	-	0.772655
kn-en	17.93	-	**24.18**	24.01	-	0.744802	**0.751223**
ml-en	19.52	-	**22.84**	22.1	-	**0.745908**	0.744459
ta-en	**23.62**	-	22.75	21.37	-	**0.74509**	0.742311
te-en	19.89	-	**24.02**	22.37	-	**0.745885**	0.743435

Table 2: Results: *XX-en* is the translation direction. *IA, DR, All* are Indo-Aryan, Dravidian and All Indic languages respectively. The numbers under BLEU and AMFM headings represent BLEU score and AMFM score respectively.

the evaluation of MultiIndicMT task[6]. Tokenization is performed using moses-tokenizer (Koehn et al., 2007). For IA-En, DR-En, and All-En, we report results provided by the organizers. Table 2 also reports the Adequacy-Fluency Metrics (AM-FM) for Machine Translation (MT) Evaluation (Banchs et al., 2015) provided by organizers.

[6] http://lotus.kuee.kyoto-u.ac.jp/WAT/evaluation/automatic_evaluation_systems/automaticEvaluationEN.html

The BLEU score in table 2 highlights that the multilingual model outperforms the simpler bilingual models. Although we did not submit bilingual models in the shared task submission, we use it here as a baseline to compare with multilingual models. Moreover, upon grouping languages based on their language families, significant improvement in BLEU scores is observed due to less confusion and better learning of the language representations in shared encoder-decoder architecture. We ob-

lang1 \ lang2	bn	gu	hi	mr	or	pa	kn	ml	ta	te
bn	-	37.86	80.63	55.1	34.81	35.93	24.69	54.83	61.79	60.89
gu	70.47	-	93.51	83.52	51.02	54.09	49.22	61.21	46.85	71.74
hi	68.96	42.97	-	59.62	30.79	38.29	27.66	52.68	55.77	60.5
mr	72.35	58.91	91.53	-	40.36	45.2	38.04	60.91	53.59	69.23
or	83.6	65.83	86.47	73.82	-	48.94	48.1	61.66	44.71	68.7
pa	72.39	58.54	90.19	69.36	41.05	-	36.64	60.16	59.18	68.58
kn	63.08	67.57	82.64	74.04	51.17	46.48	-	74.39	50.34	84.07
ml	67.37	40.4	75.68	56.99	31.54	36.69	35.77	-	66.00	68.86
ta	63.49	25.86	67.00	41.94	19.13	30.19	20.24	55.19	-	56.59
te	71.66	45.36	83.26	62.05	33.67	40.07	38.72	65.96	64.82	-

Table 3: Shared Vocabulary: Percentage of vocabulary (after applying BPE) of lang1 present in lang2 (rows: lang1, columns: lang2) after transliteration to a common script (devnagari)

serve that the BLEU score increases by 14 percent on average when the languages are grouped based on their families *(IA-En & DR-En)* and by 7 percent when all languages are combined in a single multilingual model *(All-En)* as compared to the bilingual models. The *IA-En* and *DR-En* BLEU scores being better than both bilingual and multilingual *(All-En)* models encourage the exploitation of linguistic insights like languages relatedness and lexical closeness among language families.

Table 3 shows the percentage of vocabulary overlap in two languages. We get the vocabulary of each language using the source language part of the BPE processed parallel train set files as used in *All-En* experiment. The vocabulary size for each language is different. Equation 1 states how the value in each cell is calculated. $V1$, $V2$ are the vocabularies of lang1 & lang2 respectively. The numerator is the count of intersection of the two vocabularies and denominator is the count of the vocabulary of lang1.

$$\frac{|V1 \cap V2|}{|V1|} * 100 \quad (1)$$

Almost all indic languages provided in the task *bn, gu, (hi,mr), or, pa, kn, ml, ta, te,* use different scripts except *hi* and *mr*. Both *hi* and *mr* utilize the same script (devnagari). It is clear from Table 3 that transliteration to a common script helps in increasing the shared vocabulary and helps the model to leverage the benefit of the lexical similarity between languages.

5 Conclusion

In this paper, we study the influence of sharing encoder-decoder parameters between related languages in multilingual NMT by performing experiments with the grouping of languages based on language family. Furthermore, we also perform experiments of multilingual NMT with all Indic language data converted to the same script, which helps the model in learning better translation by utilizing the benefit of better shared vocabulary. In the future, we plan to utilize monolingual data from (Kakwani et al., 2020) to improve multilingual NMT further.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2018. Unsupervised neural machine translation. *ArXiv*, abs/1710.11041.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Rafael E. Banchs, Luis F. D'Haro, and Haizhou Li. 2015. Adequacy–fluency metrics: Evaluating mt in the continuous space model framework. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(3):472–482.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with

subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Yong Cheng, Qian Yang, Yang Liu, Maosong Sun, and Wei Xu. 2017. Joint training for pivot-based neural machine translation. In *IJCAI*, pages 3974–3980.

Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan. 2020. A survey of multilingual neural machine translation. *ACM Computing Surveys (CSUR)*, 53(5):1–38.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 866–875.

Yarin Gal and Zoubin Ghahramani. 2016. A theoretically grounded application of dropout in recurrent neural networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS, pages 1027–1035, Red Hook, NY, USA. Curran Associates Inc.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning.

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. 2017a. Google's multilingual neural machine translation system: Enabling zero-shot translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

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

Divyanshu Kakwani, Anoop Kunchukuttan, Satish Golla, NC Gokul, Avik Bhattacharyya, Mitesh M Khapra, and Pratyush Kumar. 2020. inlpsuite: Monolingual corpora, evaluation benchmarks and pre-trained multilingual language models for indian languages. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 4948–4961.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 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.

Anoop Kunchukuttan and Pushpak Bhattacharyya. 2020. Utilizing language relatedness to improve machine translation: A case study on languages of the indian subcontinent. *CoRR*, abs/2003.08925.

Surafel M. Lakew, Quintino F. Lotito, Matteo Negri, Marco Turchi, and Marcello Federico. 2018. Improving zero-shot translation of low-resource languages.

Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *Advances in Neural Information Processing Systems (NeurIPS)*.

Guillaume Lample, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. In *International Conference on Learning Representations*.

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

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

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.

Gowtham Ramesh, Sumanth Doddapaneni, Aravinth Bheemaraj, Mayank Jobanputra, Raghavan AK, Ajitesh Sharma, Sujit Sahoo, Harshita Diddee, Divyanshu Kakwani, Navneet Kumar, et al. 2021.

Samanantar: The largest publicly available parallel corpora collection for 11 indic languages. *arXiv preprint arXiv:2104.05596*.

Rico Sennrich, B. Haddow, and Alexandra Birch. 2016a. Neural machine translation of rare words with subword units. *ArXiv*, abs/1508.07909.

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

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2019. Mass: Masked sequence to sequence pre-training for language generation. In *International Conference on Machine Learning*, pages 5926–5936.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Y. Wu, M. Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, J. Klingner, Apurva Shah, M. Johnson, X. Liu, Lukasz Kaiser, Stephan Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, George Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *ArXiv*, abs/1604.02201.