Multilingual Multi-Domain NMT for Indian Languages

Sourav Kumar
IIIT Hyderabad
sourav.kumar@research.iiit.ac.in

Salil Aggarwal
IIIT Hyderabad
salil.aggarwal@research.iiit.ac.in

Dipt Misra
IIIT Hyderabad
dipti@iiit.ac.in

Abstract

India is known as the land of many tongues and dialects. Neural machine translation (NMT) is the current state-of-the-art approach for machine translation (MT) but performs better only with large datasets which Indian languages usually lack, making this approach infeasible. So, in this paper, we address the problem of data scarcity by efficiently training multilingual and multilingual multi domain NMT systems involving languages of the **Indian subcontinent**. We are proposing the technique for using the joint domain and language tags in a multilingual setup. We draw three major conclusions from our experiments: (i) Training a multilingual system via exploiting lexical similarity based on language family helps in achieving an overall average improvement of **3.25 BLEU points** over bilingual baselines, (ii) Technique of incorporating domain information into the language tokens helps multilingual multi-domain system in getting a significant average improvement of **6 BLEU points** over the baselines, (iii) Multistage fine-tuning further helps in getting an improvement of **1-1.5 BLEU points** for the language pair of interest.

1 Introduction

Good translation systems are an important requirement due to substantial government, business and social communication among people speaking different languages. **Neural machine translation** (Vaswani et al., 2017) is the current state-of-the-art approach for machine translation in both academia and industry (Bahdanau et al., 2014). The success of NMT heavily relies on a huge amount of parallel sentences as training data. But using the traditional approaches (Vaswani et al., 2017), one would still need to train a separate model for each translation direction and also a lot of parallel human translated corpora which is again expensive to generate. But on the other hand, **multilingual neural machine translation** (Johnson et al., 2017) enables training a single model that supports translation from multiple source languages to a single target language or from a single source language to multiple target languages. In addition to this, another benefit of a training a single model for multiple translation directions is the ability to learn not just from the training data of the language pair of interest, but also from other language pairs. But this learning is hindered in case of language pairs that do not show any kind of relatedness among themselves. But on the other hand, Indian languages exhibit a lot of lexical and structural similarities on account of sharing a common ancestry. It is therefore important to utilize the lexical similarity (Kunchukuttan and Bhattacharyya, 2020) of these languages to build efficient systems by combining all the related languages.

Also, in a typical in-domain MT scenario, the amount of parallel texts from a single domain is not enough to train a good translation system, even for multilingual systems. Apart from this, one has to train an individual MNMT system for each domain. So we propose a technique for creating efficient multilingual multi-domain NMT systems which help in overcoming the above mentioned limitations. In this work, we treat different domains as distinct languages: for example, instead of Hindi-English translation we see it as translating Hindi-health to English-health. We utilized our multilingual NMT approach in a multi-domain setting and our results confirm that our multilingual multi-domain system significantly outperforms in-domain baselines as well as it also give improvement for out-of-domain translations. The paper is organized as follows: Section 2 talks about the related work. Methodology for our

Proceedings of Recent Advances in Natural Language Processing, pages 727–733
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_083

experiments is explained in Section 3, followed by experimental details and results in Section 4 and Section 5 respectively. All the conclusions and the future work have been briefly discussed in Section 6.

2 Related Work

Due to simplicity, generality and effectiveness of Neural Machine Translation (NMT), it has become the most prominent approach to machine translation (Luong et al., 2015; Bahdanau et al., 2014; Johnson et al., 2017; Wu et al., 2018; Vaswani et al., 2017). Basic training procedure of NMT does not work well with only a handful of bilingual data (Koehn and Knowles, 2017), while collecting bilingual resources is arduous for Indian languages. A lot of experiments have been made to improve the quality of translation mainly including exploiting monolingual data range from back translation (Sennrich et al., 2015), dual learning (Xia et al., 2016) to Unsupervised MT (Artetxe et al., 2017; Lample et al., 2017). On the other hand, many tried to exploit parallel data of other high resource languages (Zoph et al., 2016; Firat et al., 2017; Johnson et al., 2017; Kocmi and Bojar, 2018) to either pre-train the network or jointly learn the representation.

Recently multilingual NMT has drawn more attention by several research groups. For instance, Firat et al. (2016) modify the current state-of-the-art attention NMT approach by introducing a many-to-many system, which still relied upon separate encoders and decoders for each language along with a shared attention mechanism. In contrast, Johnson et al. (2017) and Ha et al. (2016) both introduce a simple method for training a single-model multilingual NMT system, which does not require any modifications to the NMT encoder-decoder architecture of the system. The main difference is that Johnson et al. (2017) added target language identifying token in the beginning of each source sentence of the training data and Ha et al. (2016) added a language identifying token to each subword unit and apply this pre-processing to both source and target sentences of the training data. Both aims at exploiting many different languages rather than focusing on language relatedness and observes that only the many-to-one paradigm can achieve better translation results than the individually trained models. For the other two paradigms, there are various degrees of quality degradation. Also Vertan and von Hahn (2013) has put some efforts in tackling efficient NMT system in low-resource settings by considering language relatedness.

Apart from this, researchers have also explored the area of domain adaptation for NMT (Chu and Wang, 2018) and reported significant improvements. (Tars and Fishel, 2018; Kobus et al., 2016) explored multi-domain Neural machine translation for single language by adding the domain token to the input sentence.
So, we put our efforts in exploring the system performance when multilingualism is combined with multi-domain systems for major Indian languages.

3 Methodology

India is a land of diverse languages. It has many languages on the basis of regional diversities, and mainly divided into Indo-Aryan and Dravidian families. A universal characteristic of Indian languages is their complex morphology. Indian languages depict unique characteristics following default sentence structure as **subject object verb (SOV)** and relatively free word order. These languages also share many common words which have the same root and meaning. However they use different scripts derived from the ancient Brahmi script (Kunchukuttan and Bhattacharyya, 2020), but correspondences can be established between equivalent characters across scripts. So, we exploited lexical similarity for efficient MNMT. Also, in order to incorporate multiple domains into our multilingual system, we also introduced the technique of representing the domain as a new language.

3.1 Exploiting Lexical Similarity

Unlike the original multilingual NMT (Johnson et al., 2017) which aims at exploiting many different languages rather than focusing on language similarity. Thus, to exploit the language relatedness we efficiently combined the two different approaches namely **Unified Transliteration** and **Subword Segmentation** to ensure that there is a sufficient overlap between the vocabularies of the related languages.

3.1.1 Unified Transliteration

Since the languages involved in the models have different orthographies and relatedness among each

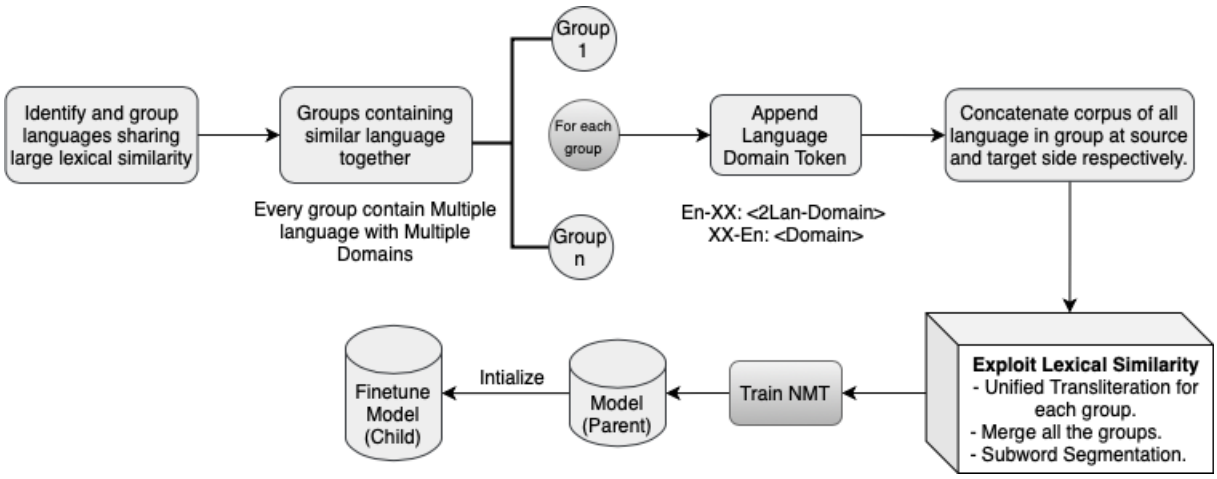

Figure 1: Multilingual Multi-Domain NMT Pipeline

other, also Indo-Aryan and Dravidian family languages don't share many common characteristics, thus in the data processing one can map these two different language families into two different common orthographies. In order to achieve this, we transliterated all the Indian language into a common script based upon their family groups using the Indic NLP library (Kunchukuttan, 2020). Indo-Aryan languages were transliterated to Hindi (Devnagri script) while Dravidian languages were transliterated to Tamil (Abugida script) to share the same surface form within their family. This unified transliteration is a string homomorphism, replacing characters in all the languages to the desired script.

3.1.2 Subword Segmentation

Most of the Indian languages are derived from the common ancient Brahmi script and share many common words at the root level. To do so, we used **Byte Pair Encoding (BPE)** (Sennrich et al., 2015) to break words into subwords. Also, BPE merge rules not only find the common subwords between two related languages but it also ensures consistency of segmentation among each considered language pair. We are learning the BPE rules by combining all the languages on the source and the target side respectively, thus further applying these rules for segmenting the corpora. This finally results in increasing the vocabularies overlap among the languages that we made share the same surface form by transliterating into a common desired script.

3.2 Multilingual and Multi-Domain Systems

Multilingual model enables us to translate to and from multiple languages using a shared word piece vocabulary, which is significantly simpler than training a different model for each language pair. Johnson et al. (2017) introduced a "language flag" based approach that shares the attention mechanism and a single encoder-decoder network to enable multilingual models. A language flag or token is part of the input sequence to indicate which direction to translate to. The decoder learns to generate the target given this input. This approach has been shown to be simple, effective and forces the model to generalize across language boundaries during training. It is also observed that when language pairs with little available data and language pairs with abundant data are mixed into a single model, translation quality on the low resource language pair is significantly improved.

Multi-domain model is a single model that supports multiple domains in one model and also allows switching between the domains when translating. Similar to Johnson et al. (2017), Tars and Fishel (2018) explored Multi-Domain Neural Machine Translation for single language by adding the domain token to the input sentence instead of the language token.

3.3 Multilingual Multi-Domain Systems

A lot of research areas have been explored separately for multilingual as well as Multi domain systems. But no work has been done on combining both these approaches for the Indian languages. To the best of our knowledge, this is the first time efforts have been made in combining these two techniques for the NMT systems trained for Indian languages. So in this paper, we present our

Data	En-hi	En-pa	En-gu	En-mr	En-bn	En-or	En-kn	En-ml	En-ta	En-te
PMI	50349	28294	41578	28794	23306	31966	28901	26916	32638	33380
ILCI	44601 (Except or and kn; Include ko)									

Table 1: Training Dataset Statistics

technique of training a multilingual multi-domain system using the same traditional encoder-decoder architecture with shared attention mechanism. To do so, we introduced a **special token** technique which incorporates the knowledge of the language as well as the domain. The core idea is to treat *'domains as distinct languages'* while training multilingual multi-domain systems. The pipeline for our technique can be seen in **Figure 1**.

For efficiently exploiting the language relatedness, we are first identifying and grouping the languages based on their lexical similarities with each other. In our case of Indian languages, all of the languages were first divided into 2 groups namely Indo Aryan and Dravidian. For each group, we appended our special token ⟨**2Lan-Domain**⟩ for one to many systems and ⟨**Domain**⟩ token for many to one systems. Then each group is being transliterated to a common script. For our purpose, Indo aryan languages were transliterated into Hindi while Dravidian languages were transliterated into Tamil. Then to increase vocabulary overlap amongst related languages at root word level, we are using BPE as discussed in **Section 3.2**. Further, we used this model learning for multistage fine tuning.

3.3.1 Multistage Fine Tuning

In the normal transfer learning Zoph et al. (2016) approach for NMT, the parent model is trained on a single high-resource language pair which may or may not be related to the child language pair of interest. To the best of our knowledge, previous transfer learning approaches for Indian languages do not exploit parallel data from multiple languages and domains. However, learning from multiple languages and domains can result in better knowledge transfer. Therefore, in this work, we propose a new transfer learning approach called as *'Multistage Transfer Learning'* to enable the low-resource languages to efficiently learn from multiple related languages as well as domains which may or may not be high-resourced. In this approach, the parent model is our multilingual multi-domain NMT system and after pre training the parent model, the child model is initialized with parent model parameters and is then fine-tuned multiple times.

The proposed approach delivers better results than multilingual multi-domain NMT because adding more languages and domains into one model may result in better knowledge transfer but it can also result in ambiguities between different languages and domains at the inference time. Accordingly, a multilingual multi-domain NMT system fine-tuned can potentially remove all the inconsistencies at the inference time. For the scope of this paper, we have performed multistage fine tuning in three different scenarios : (i) single domain multiple language, (ii) multiple domain single language and (iii) single domain single language.

4 Experimental Settings

4.1 Dataset

In our experiments, we are using the multi parallel corpus of two completely different domains namely PMI (Prime Minister of India) (Haddow and Kirefu, 2020) which contains the news domain aligned sentences and ILCI (Indian Language Corpora Initiative) (Jha, 2010) which is a combination of health as well as tourism domain sentences. ILCI corpus contains translation of the same sentence in every language pair but PMI contains roughly 60% of common sentences being translated to all language pairs, where Hindi being the high resource language.

4.2 Data Preprocessing

We also noticed the ILCI corpus contains a lot of misalignments and empty translations, so we put our efforts in cleaning the entire corpus maintaining multi parallelism. We ended up in removing 3099 sentences from the corpus. Training data set statistics of both the data sets are mentioned in **Table 1**. All of our experiments were tested on 1870 sentences from ILCI and 2390 from the PMI corpus with validation data of 500 from both. We also made sure that there is no overlap between the

Type	En-hi	En-pa	En-gu	En-mr	En-bn	En-or	En-kn	En-ml	En-ta	En-te
Bilingual Baseline	23.21	18.26	15.46	7.07	5.25	8.32	8.67	4.63	5.32	6.12
Mutli Domain Bilingual Baseline	24.48	19.41	16.23	8.37	6.36	-	-	4.86	6.09	6.75
Multilingual	28.01	26.14	20.80	13.51	10.45	14.71	15.02	9.05	9.26	8.25
Multilingual Multi Domain	28.45	26.30	21.06	14.53	10.88	14.98	15.37	9.92	9.83	7.93
Multilingual Multi Domain with Token	**29.29**	**27.29**	**21.86**	**14.57**	**11.04**	**15.53**	**15.79**	**10.16**	**9.88**	**8.67**

Table 2: PMI results (En-XX)

Type	En-hi	En-ur	En-pa	En-gu	En-mr	En-bn	En-ko	En-ml	En-ta	En-te
Bilingual Baseline	22.65	21.01	18.60	14.45	10.08	12.15	8.28	3.29	1.97	5.06
Multi Domain Bilingual Baseline	23.64	-	19.52	16.48	10.95	13.79	-	4.42	3.04	6.23
Multilingual	27.02	22.32	23.08	21.5	14.08	16.79	14.41	6.19	5.18	9.36
Multilingual Multi Domain	28.06	21.37	23.19	22.51	15.26	17.14	14.52	5.80	5.59	9.88
Multilingual Multi Domain with Token	**29.26**	**26.47**	**24.66**	**23.23**	**15.34**	**18.12**	**15.46**	**6.72**	**5.87**	**10.58**

Table 3: ILCI results (En-XX)

test and training set of PMI corpus. We used the Moses (Koehn et al., 2007) toolkit for tokenization and cleaning the English side of the data and we used the Indic NLP library (Kunchukuttan, 2020) for the normalization, tokenization and transliteration for the Indian languages. In all cases, we used BPE segmentation with 12k merge operations as described in **Section 3.1.2**

4.3 Training and Evaluation Details

For all of our experiments, we use the **OpenNMT-py** (Klein et al., 2017) toolkit. We used the Transformer model with 6 layers in both the encoder and decoder, each with 512 hidden units. The word embedding size is set to 512 with 8 heads. The training is done in batches of maximum 4096 tokens at a time with dropout set to 0.3. We use the Adam (Kingma and Ba, 2014) optimizer to optimize model parameters. We validate the model every 5,000 steps via BLEU (Papineni et al., 2002) and perplexity on the development set. We are training all of our NMT models with early stopping criteria based on validation set accuracy. During testing, we rejoin the translated BPE segments and convert the translated sentences back to their original language scripts. Finally, we evaluate the accuracy of our translation models using BLEU.

5 Results and Analysis

We report the results of bilingual baseline, multi-domain bilingual baseline, multilingual, multilingual multi-domain and multilingual multi-domain with special tokens for both the translation directions, XX-En and En-XX (where XX denotes Indian Languages). Later, we also compared the results of multistage fine tuning with the above experiments for one language from each family.

Table 2 and 3 shows our main results for English to Indian languages (En-XX) translation direction for PMI and ILCI corpus respectively. In both the cases, we observed that our multilingual model shows significant improvements over the baseline, increasing average BLEU score of 5 and 4 respectively. The reason behind this is that in the En-XX direction, language flags are used on the source side which then helps the decoder to identify the direction it translates to.

Table 4 and 5 shows our main results for Indian languages to English (XX-En) translation direction for PMI and ILCI corpus respectively. In the case of the ILCI dataset, we do not observe any significant improvements. The reason for this might be the multi parallel nature of the ILCI dataset where each English sentence on the target side appears multiple times in the model, thereby creating ambiguities in the model. But for the case of the PMI dataset, we observed an average improvement of 6 BLEU points, mainly due to large improvements in low resource languages. The reason for the increase in BLEU score for PMI is that the distribution of data is not uniform. Some of the languages in PMI corpus are low-resource as compared to others thus allowing other high resource languages to assist in the learning process for low resource languages thus removing the ambiguity. We also showed that the results of the multilingual multi-domain system with our special language domain outperforms the without token case for both the domains giving

Type	Hi-en	Pa-en	Gu-en	Mr-en	Bn-en	Or-en	Kn-en	Ml-en	Ta-en	Te-en
Bilingual Baseline	24.69	19.80	20.16	11.70	10.25	13.80	13.32	11.30	9.82	13.39
Multi Domain Bilingual Baseline	25.78	21.09	22.75	13.22	12.11	-	-	13.63	11.08	14.45
Multilingual	26.63	24.41	24.11	19.62	17.37	20.28	21.14	19.01	18.44	19.91
Multilingual Multi Domain	29.54	27.12	26.00	21.30	18.63	21.46	22.40	19.91	20.04	20.81
Multilingual Multi Domain with Token	**31.04**	**28.89**	**28.39**	**22.92**	**19.46**	**22.94**	**23.95**	**21.48**	**21.01**	**22.24**

Table 4: PMI results (XX-En)

Type	Hi-en	Ur-en	Pa-en	Gu-en	Mr-en	Bn-en	Ko-en	Ml-en	Ta-en	Te-en
Bilingual Baseline	24.08	20.14	21.14	18.10	15.46	14.87	12.65	7.51	5.11	10.81
Multi Domain Bilingual Baseline	25.45	-	22.53	19.34	17.00	16.11	-	9.78	6.65	12.83
Multilingual	18.40	15.52	16.59	16.98	15.67	13.93	13.99	10.33	8.35	12.65
Multilingual Multi Domain	19.41	15.80	17.09	17.30	15.85	14.32	14.29	10.51	8.93	13.42
Multilingual Multi Domain with Token	**23.21**	**18.42**	**20.62**	**21.04**	**19.52**	**17.61**	**16.85**	**12.45**	**10.09**	**15.55**

Table 5: ILCI results (XX-en)

an average improvement of **1.5 Bleu points** in the En-XX direction and **4 BLEU points** in XX-En direction over a normal multilingual system. We also experimented the Multistage Fine tuning for Punjabi and Tamil following the above three different scenarios mentioned above in **Section 3.3.1** and observed an improvement within **1 - 1.5 BLEU points** over the multilingual multi-domain system.

6 Conclusions and Future Work

In this paper, we explored different effective methods to exploit parallel data from multiple related languages and domains to improve the translation between Indian languages and English. Our results show that the multilingual models accuracy depends upon the type of dataset in hand. As we observed in the case of PMI and ILCI, multilingual models trained on the PMI dataset increased our average BLEU score while the model trained on the ILCI dataset decreased the BLEU score due to increase in ambiguity. We then introduced multilingual multi-domain models and observed that this idea helps in removing the ambiguity we faced in the multilingual system using multi parallel data for training thus improving the translation quality by showing improvements in BLEU scores. In this work, we also introduced a new technique of adding a domain as a separate language by modifying the language token to language domain token. Our experiments also confirm that this new technique always outperforms all the models we discussed above. At last, we also explored the concept of Multistage Fine Tuning in which we transfer the learning of the parent model to the child in multiple stages. In future, we would like to work on effective techniques to exploit monolingual data and parallel data from other languages together to improve the translation quality. Also, we will try to generalize this idea of exploiting the related languages to other NLP related applications like sentiment analysis.

References

Mikel Artetxe, Gorka Labaka, Eneko Agirre, and Kyunghyun Cho. 2017. Unsupervised neural machine translation. *arXiv preprint arXiv:1710.11041*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. *arXiv preprint arXiv:1806.00258*.

Orhan Firat, Kyunghyun Cho, and Yoshua Bengio. 2016. Multi-way, multilingual neural machine translation with a shared attention mechanism. *arXiv preprint arXiv:1601.01073*.

Orhan Firat, Kyunghyun Cho, Baskaran Sankaran, Fatos T Yarman Vural, and Yoshua Bengio. 2017. Multi-way, multilingual neural machine translation. *Computer Speech & Language*, 45:236–252.

Thanh-Le Ha, Jan Niehues, and Alexander Waibel. 2016. Toward multilingual neural machine translation with universal encoder and decoder. *arXiv preprint arXiv:1611.04798*.

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

Girish Nath Jha. 2010. The tdil program and the indian langauge corpora intitiative (ilci). In *LREC*.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of ACL 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Catherine Kobus, Josep Crego, and Jean Senellart. 2016. Domain control for neural machine translation. *arXiv preprint arXiv:1612.06140*.

Tom Kocmi and Ondřej Bojar. 2018. Trivial transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1809.00357*.

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.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

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

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

Guillaume Lample, Alexis Conneau, Ludovic Denoyer, and Marc'Aurelio Ranzato. 2017. Unsupervised machine translation using monolingual corpora only. *arXiv preprint arXiv:1711.00043*.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

Sander Tars and Mark Fishel. 2018. Multi-domain neural machine translation. *arXiv preprint arXiv:1805.02282*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Cristina Vertan and Walther von Hahn. 2013. Language diversity and implications for language technology in the multilingual europe. In *Proceedings of the Workshop on Adaptation of Language Resources and Tools for Closely Related Languages and Language Variants*, pages 2–6.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018. Adversarial neural machine translation. In *Asian Conference on Machine Learning*, pages 534–549. PMLR.

Yingce Xia, Di He, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *arXiv preprint arXiv:1611.00179*.

Barret Zoph, Deniz Yuret, Jonathan May, and Kevin Knight. 2016. Transfer learning for low-resource neural machine translation. *arXiv preprint arXiv:1604.02201*.