ICON 2020

17th International Conference on Natural Language Processing

Proceedings of the Adap-MT 2020 Shared Task

December 18 - 21, 2020
Indian Institute of Technology Patna, India

Introduction

These shared task proceedings concluded the shared task on Low Resource Domain Adaptation for Indic Machine Translation, named as ADAP-MT 2020, launched on 7th October 2020. The shared task was collocated with the 17th International Conference on Natural Language Processing (ICON 2020), held at IIT-Patna, India. The goal of the shared task was to show how MT systems trained on general domains perform on Indic Languages and low resource domain adaptation using a limited domain-specific parallel corpus.

Two subtasks were part of this shared task. In the first subtask, the participants were asked to develop MT systems for the General domain. The second subtask required the development of MT systems for specified domains - AI, Chemistry utilizing general domain parallel data and very limited domain-specific data for domain adaptation. Parallel corpora of three language pairs - English - Hindi, English - Telugu and Hindi - Telugu were released for the shared task.

We received five system submissions and system description papers. Each system description paper was reviewed by two members of the reviewing committee – all papers were accepted. The submitted systems were evaluated using BLEU scores.

Statistical MT and Neural MT were the two kinds of models used by the participants. Subword level NMT models with Byte Pair Encoding with self - Attention were mostly used. Participants also augmented the training data with techniques like oversampling of domain-specific data and mixed fine-tuning. We would like to thank the ICON-2020 organizers, the shared task participants, the authors, and the reviewers for making this shared task successful.

Shared task page: `http://ssmt.iiit.ac.in/machinetranslation`
Main conference page: `https://www.iitp.ac.in/~ai-nlp-ml/icon2020/index.html`

Organizing Committee:

Dipti Misra Sharma (IIIT-Hyderabad)
Asif Ekbal (IIT-Patna)
Karunesh Arora (C-DAC, Noida)
Sudip Kumar Naskar (Jadavpur University)
Dipankar Ganguly (C-DAC, Noida)
Sobha L (AUKBC-Chennai)
Radhika Mamidi (IIIT-Hyderabad)
Sunita Arora (C-DAC, Noida)
Pruthwik Mishra (IIIT-Hyderabad)
Vandan Mujadia (IIIT-Hyderabad)

Table of Contents

Shared Task Program

Monday, December 21, 2020

+ 14:00 - 14:30 **Talk by Sobha L, AUKBC-Chennai**

+ 14:30 - 14:45 Shared Task Overview

Presentations

14:50 - 15:00 *Terminology-Aware Sentence Mining for NMT Domain Adaptation: ADAPT's Submission to the Adap-MT 2020 English-to-Hindi AI Translation Shared Task*
Rejwanul Haque, Yasmin Moslem and Andy Way

15:03 - 15:13 *AdapNMT : Neural Machine Translation with Technical Domain Adaptation for Indic Languages*
Hema Ala, Dipti Misra Sharma

15:16 - 15:26 *Domain Adaptation of NMT models for English-Hindi Machine Translation Task at AdapMT ICON 2020*
Ramchandra Joshi, Rushabh Karnavat, Kaustubh Jirapure, Raviraj Joshi

15:29 - 15:39 *JUNLP@ICON2020: Low Resourced Machine Translation for Indic Languages*
Sainik Kumar Mahata, Dipankar Das, Sivaji Bandyopadhyay

15:42 - 15:52 *MUCS@Adap-MT 2020: Low Resource Domain Adaptation for Indic Machine Translation*
Asha Hegde, H. L. Shashirekha

JUNLP@ICON2020: Low Resourced Machine Translation for Indic Languages

Sainik Kumar Mahata, Dipankar Das, Sivaji Bandyopadhyay
Computer Science and Engineering
Jadavpur University
sainik.mahata@gmail.com, dipankar.dipnil2005@gmail.com
sivaji.cse.ju@gmail.com

Abstract

In the current work, we present the description of the systems submitted to a machine translation shared task organized by ICON 2020: 17th International Conference on Natural Language Processing. The systems were developed to show the capability of general domain machine translation when translating into Indic languages, English-Hindi, in our case. The paper shows the training process and quantifies the performance of two state-of-the-art translation systems, viz., Statistical Machine Translation and Neural Machine Translation. While Statistical Machine Translation systems work better in a low-resource setting, Neural Machine Translation systems are able to generate sentences that are fluent in nature. Since both these systems have contrasting advantages, a hybrid system, incorporating both, was also developed to leverage all the strong points. The submitted systems garnered BLEU scores of 8.701943312, 0.6361336198, and 11.78873307 respectively and the scores of the hybrid system helped us to the fourth spot in the competition leaderboard.

1 Introduction

Machine **T**ranslation (MT) is the translation of one natural language to another using software. Generally, training a good translation system requires the availability of a large and good quality parallel corpus. These corpora are easily available for languages that are spoken globally and have a large digital footprint. But finding the same for less-resourced languages, that are not universally recognized and do not have a large digital presence, is a challenge. This leads to the development of translation systems that do not produce quality results. The present work aims to solve a similar issue and focuses on showing the capability of general domain machine translation when translating into Indic languages, English-Hindi, in our case. The literature includes the description and training process of state-of-the-art translation systems and finally quantifies their performance with respect to the data provided as part of a shared task organized by ICON 2020: 17th International Conference on Natural Language Processing[1].

The shared task was divided into two sub-tasks,

- **SubTask 1** : To show sentence level Machine translation capability for on General domain.
- **SubTask 2** : To show sentence level Machine translation capability for on specified domains.

We took part in the first sub-task and proceeded with developing translation systems with the help of the provided English-Hindi parallel corpus.

Using the provided parallel corpus, we developed three systems. The first two systems was based on **S**tatistical **M**achine **T**ranslation (SMT) and **N**eural **M**achine **T**ranslation (NMT). For training the SMT system, Moses Toolkit (Koehn et al., 2007) was used. The NMT system was a character based seq-to-seq model, that was trained using Bi-Directional **L**ong **S**hort-**T**erm **M**emory (LSTM) cells (Hochreiter and Schmidhuber, 1997). The third system was a hybrid system, that works on the principles of **A**utomated **P**ost **E**diting (APE). In this model, a transformer (Vaswani et al., 2017) based NMT model was used to post edit the outputs, generated by an SMT based translation system.

The rest of the paper is organized as follows. Section 2 describes the parallel corpus that was used to train the above-mentioned translation systems. Section 3 contains the description and the training processes of all the developed translation systems. This will be followed by the evaluation results and discussion in Section 4 and 5. Finally,

[1] https://ssmt.iiit.ac.in/machinetranslation.html

Proceedings of the 17th International Conference on Natural Language Processing: Adap-MT 2020 Shared Task, pages 1–5
Patna, India, December 18 - 21, 2020.

concluding remarks and future scopes have been discussed in Section 6.

2 Parallel Corpus

Multiple English-Hindi parallel corpora were provided by the organizers for training the translation systems. Among these, we decided on using the parallel corpus from CVIT-PIB[2] and CVIT-MKB[3]. Another high-quality corpus from TDIL[4]was also used to train our developed systems. The number of parallel sentences in the CVIT-MKB dataset was 5,272, in the CVIT-PIB dataset were 1,95,208, and in the TDIL dataset were 50,000. In total, we were able to arrange for parallel English-Hindi corpora of 2,50,480 sentences. The data was then tokenized to be used for our further experiments. For tokenizing the English data, NLTK[5] (Bird, 2006) was used and for tokenizing the Hindi data, Indic NLP Library[6] (Kunchukuttan, 2020) was used.

3 Machine Translation

After the English-Hindi parallel corpora were compiled, we proceeded to develop our MT systems. As discussed earlier, the first two MT systems were based on SMT and NMT. The third MT system was a hybrid system, using both SMT and NMT, based on the transformer architecture, and worked on the principle of APE. The description of the all the three systems and the training process for the same is given in Sections 3.1, 3.2 and 3.3 respectively.

3.1 Statistical Machine Translation

For designing the model we followed some standard preprocessing steps on 2,50,480 sentence pairs, which are discussed below.

3.1.1 Preprocessing

The following steps were applied to preprocess and clean the data before using it for training our Statistical machine translation model. We used the NLTK toolkit[7] for performing the steps.

- **Tokenization**: Given a character sequence and a defined document unit, tokenization is the task of chopping it up into pieces, called tokens. In our case, these tokens were words, punctuation marks, numbers. NLTK supports tokenization of Lithuanian as well as English texts.

- **Truecasing**: This refers to the process of restoring case information to badly-cased or non-cased text (Lita et al., 2003). Truecasing helps in reducing data sparsity.

- **Cleaning**: Long sentences (No. of tokens > 80) were removed.

3.1.2 Moses

Moses is a statistical machine translation system that allows you to automatically train translation models for any language pair when trained with a large collection of translated texts (parallel corpus). Once the model has been trained, an efficient search algorithm quickly finds the highest probability translation among the exponential number of choices.

We trained Moses using 2,50,480 sentence pairs provided by the organizers, with English as the source language and Hindi as the target language. For building the Language Model we used KenLM[8] (Heafield, 2011) with 3-grams from the target corpus.

Training the Moses statistical MT system resulted in the generation of the Phrase Model and Translation Model that helps in translating between source-target language pairs. Moses scores the phrase in the phrase table with respect to a given source sentence and produces the best-scored phrases as output.

3.2 Neural Machine Translation

In order to develop the NMT framework, we decided to employ a character-level neural machine translation system.

The **C**haracter **b**ased **NMT** (CNMT) is based on the architecture as described in Lee et al. (2017) and it relies on the sequence-to-sequence (Sutskever et al., 2014) model. We opted for character embedding based NMT for this task because of the benefits it provides over word embedding based NMT. The benefits, as stated in Chung et al. (2016), are

- capability to model morphological variants

- overcomes out-of-vocabulary issue

[2]http://preon.iiit.ac.in/ jerin/resources/datasets/pib_v0.2.tar
[3]http://preon.iiit.ac.in/ jerin/resources/datasets/mkb-v0.tar
[4]https://tdil.meity.gov.in/
[5]https://www.nltk.org/
[6]https://github.com/anoopkunchukuttan/indic_nlp_library
[7]https://www.nltk.org/
[8]https://kheafield.com/code/kenlm/

- do not require segmentation

The seq2seq model takes a sequence $X = x_1, x_2, ..., x_n$ as input and tries to generate the target sequence $Y = y_1, y_2, ..., y_m$ as output, where x_i and y_i are the input and target symbols, respectively. The architecture of seq2seq model comprises of two parts, the encoder and decoder.

In order to build the encoder, we used four bidirectional layers of LSTM cells. The input of the cell was one hot tensor of English sentences (encoding at the character level). The internal states of each cell were preserved and the outputs were discarded. The purpose of this is to preserve the information at the context level. These states were then passed on to the decoder cell as initial states.

For building the decoder, again two layers of LSTM cell were used with hidden states from the encoder as initial states. It was designed to return both sequences and states. The input to the decoder was one hot tensor (embedding at character level) of Hindi sentences while the target data was identical, but with an offset of one time-step ahead. The information for generation is gathered from the initial states passed on by the encoder. Thus, the decoder learns to generate target data [t+1,...] given targets [..., t] conditioned on the input sequence. It essentially predicts the output sequence, one character per time step.

For training the model, batch size was set to 64, number of epochs was set to 100, activation function was softmax, optimizer chosen was nadam and loss function used was sparse categorical cross-entropy. Learning rate was set to 0.001. The overall architecture is shown in Figure 1.

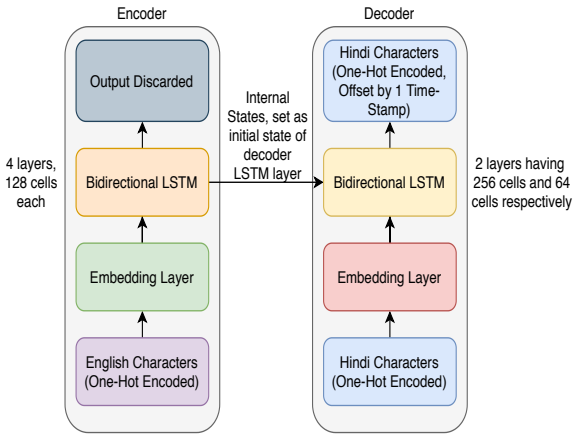

Figure 1: Character based Neural Machine Translation Architecture.

3.3 Hybrid Translation System

The NMT system used for the hybrid translation system is based on the transformer architecture. RNNs typically read one word at a time and perform multiple operations before generating output. But it has been illustrated that the more the number of steps, the harder it is for the network to learn how to make decisions (Bahdanau et al., 2014). Parallelly, RNNs are sequential, and hence taking advantage of parallel computing offered by state-of-the-art computing devices is very difficult.

On the contrary, Transformer models rely heavily on self-attention, thus eliminating the concept of recurrence found in RNN based architectures. In its absence, a positional encoding is added to the input and outputs to mimic the idea of time-steps in a recurrent network. A Transformer model comprises two parts, an encoder, and a decoder, where the encoder is composed of uniform layers, each built of two sublayers; a multi-head self-attention layer, and a position-wise feed-forward network layer. Instead of computing single attention, this stage computes multiple attention blocks over the source, concatenates them, and projects them onto space with the initial dimensionality. On the other side, the feed-forward network sub-layer is a fully connected network used to process the attention sublayers, by applying two linear transformations on each position and a ReLU activation (Vaswani et al., 2017).

The decoder operates similarly, but generates one word at a time, from left to right. The first two steps are similar to the encoder and attend only to past words. The third stage is multi-head attention that attends to these past words, in addition to the final representations generated by the encoder. The fourth stage constitutes another position-wise feed-forward network. Finally, a softmax layer allows the mapping of target word scores into target words. Figure 2 shows the architecture of NMT based on transformer architecture.

For the hybrid model, we intended to merge the SMT and NMT architectures as both these models have their own advantages. So, to incorporate the advantages of both these models into a single system, we decided to merge them in a way that is similar to the APE architecture. For this, we divided the compiled parallel corpus into two parts, one containing 1,50,480 sentences and the other containing 1,00,000 parallel sentences. The first parallel corpus was used to train an SMT sys-

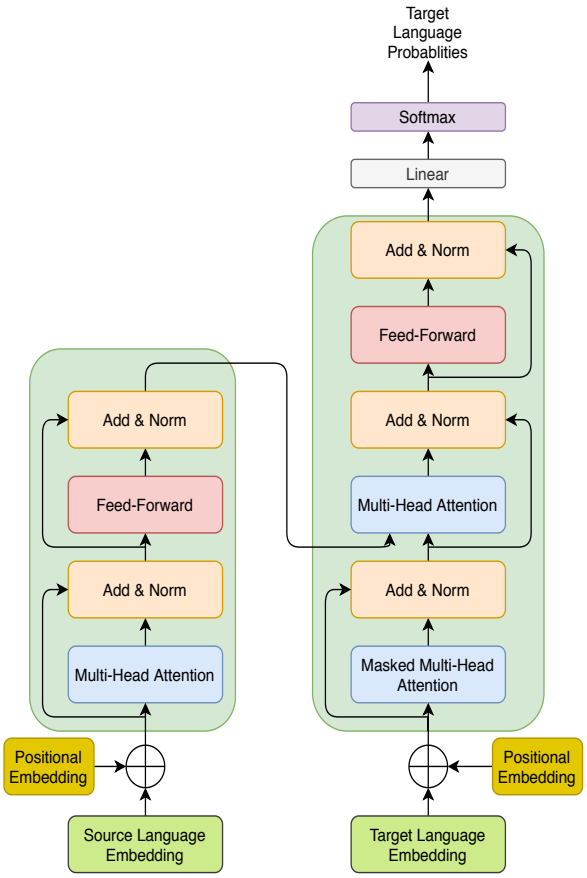

Figure 2: NMT based on Transformer Architecture.

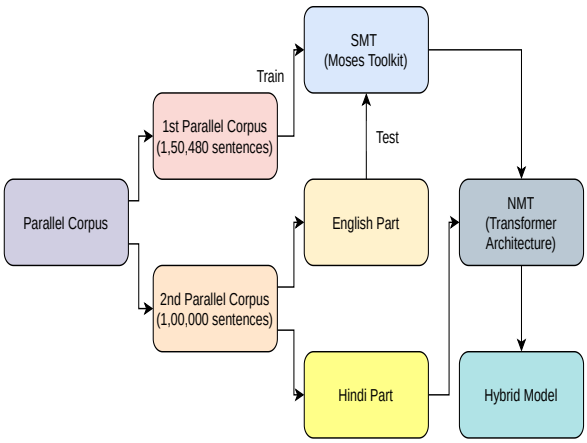

Figure 3: Architecture of the Hybrid System.

System	BLEU
SMT	8.701943312
NMT	0.6361336198
Hybrid System	11.78873307

Table 1: Evaluation of the submitted systems.

tem, built using Moses Toolkit. This was done because SMT architectures tend to work well in a low-resource setting. After training the SMT system, the second parallel corpus was used to tune the model. For this, we fed the SMT system with the English part of the second parallel corpus. In turn, the SMT model gave us the translation of these sentences as output. These outputs were then considered as source sentences to an NMT model and the Hindi part of the second parallel corpus was considered as the target. The architecture of the hybrid model is shown in Figure 3.

4 Evaluation

For evaluation purposes, the organizers provided us with a test data of 507 sentences. Upon evaluation, the performance of our systems was calculated using BLEU (Papineni et al., 2002) metric and they are shown in Table 1.

5 Discussion

From Table 1, we can see that SMT performs very well when participating languages belong to a low-resourced setting (Banerjee et al., 2018; Koehn and Knowles, 2017). This is due to the fact that the training data provided by the organizers was small and hence, belonged to similar domains. In general, SMT systems have a higher output quality when trained using domain specific training data since the texts belonging to same domain follow same pattern or usage of words. Also we can see that, during the usage of character based NMT systems, the quality of the output drops drastically. This happens as NMT systems tend to work better when there is a significant overlap between the character set of the participating source and the target languages. Due to the same reason, we see a significant increase in the performance of the hybrid system. This happens, as the second NMT system, that was based on the transformer architecture, is fed with Hindi sentences and learns to map it to Hindi sentences again, during the training process. Hence, there is a significant overlap between the vocabulary sets and hence the increase in performance.

6 Conclusion

The present paper describes the systems submitted to the translation shared task organized by ICON 2020: 17th International Conference on Natural Language Processing. We participated in the English-Hindi translation task and the training data belonged to the general domain. Three systems,

SMT, NMT, and a hybrid model was trained using these data. The models were pretty straightforward and did not contain any recent research advancements in the field of Machine Translation. As a future prospect, we would like to experiment with Transfer Learning methods, that learn from large data, and incorporate the knowledge onto models, trained using fewer data. This would be a good option as all the language options of the shared task were Indic languages and good quality and robust multi-lingual translation system can be built out of it.

Acknowledgement

This work is supported by Digital India Corporation, MeitY, Government of India, under the Visvesvaraya PhD for Electronics & IT

References

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

Tamali Banerjee, Anoop Kunchukuttan, and Pushpak Bhattacharya. 2018. Multilingual indian language translation system at wat 2018: Many-to-one phrase-based smt. In *WAT@ PACLIC*.

Steven Bird. 2006. NLTK: The Natural Language Toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, Sydney, Australia. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. *arXiv preprint arXiv:1603.06147*.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej 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. Association for Computational Linguistics.

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.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully character-level neural machine translation without explicit segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. Truecasing. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 152–159. Association for Computational Linguistics.

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

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

AdapNMT : Neural Machine Translation with Technical Domain Adaptation for Indic Languages

Hema Ala
LTRC, IIIT-Hyderabad, India
hema.ala@research.iiit.ac.in

Dipti Misra Sharma
LTRC, IIIT-Hyderabad, India
dipti@iiit.ac.in

Abstract

Adapting new domain is highly challenging task for Neural Machine Translation (NMT). In this paper we show the capability of general domain machine translation when translating into Indic languages (English - Hindi and Hindi - Telugu), and low resource domain adaptation of MT systems using existing general parallel data and small in domain parallel data for AI and Chemistry Domains. We carried out our experiments using Byte Pair Encoding(BPE) as it solves rare word problems. It has been observed that with addition of little amount of in-domain data to the general data improves the BLEU score significantly.

1 Introduction

Due to the fact that Neural Machine Translation (NMT) is performing better compared to the traditional statistical machine translation (SMT) models, it has become very popular in the recent years. NMT systems require a large amount of training data and thus perform poorly relative to phrase-based machine translation (PBMT) systems in low resource and domain adaptation scenarios (Koehn and Knowles, 2017). One of the challenges in NMT is domain adaptation, it becomes more challenging when it comes to low resource Indic languages and technical domains like Artificial Intelligence(AI) and Chemistry as these domains may contain many technical terms and equations etc. In a typical domain adaptation setup like ours, we have a large amount of out-of-domain bilingual training data for which we need to train a NMT model, we can treat this as a baseline model. Now given only an additional small amount of in-domain data, the challenge is to improve the translation performance on the new domain. Domain adaptation became very popular in these times, but very few works have been carried out on technical domains like chemistry, computer science, etc. Therefore we adopted two new technical domains in our experiments, those include Artificial Intelligence and Chemistry provided by ICON Adap-MT 2020 shared task for English - Hindi and Hindi - Telugu language pairs. In our approach first we train a general models(baseline models) which trains based on only general data, we test domain data (AI, Chemistry) on this general model then we try to improve performance of this new domain by training another model which uses combined training data(general data + domain data). Inspired from (Sennrich et al., 2015) , we encode rare and unknown words as sequences of sub word units using Byte Pair Encodings(BPE) in order to make our NMT model capable of open vocabulary translation, this is further discussed in 3.2.

2 Background & Motivation

Domain Adaptation has became an active research topic in NMT. Freitag and Al-Onaizan (2016) proposed two approaches, continue the training of the baseline model(general model) only on the in-domain data (domain data) and ensemble the continue model with the baseline model at decoding time. Zeng et al. (2019) proposed iterative dual domain adaptation framework for NMT, which continuously fully exploits the mutual complementarity between in-domain and out-domain corpora for translation knowledge transfer. Apart from these domain adaptation techniques, there exists some approaches which has domain terminology and how to use that in NMT. Similarly Hasler et al.

Proceedings of the 17th International Conference on Natural Language Processing: Adap-MT 2020 Shared Task, pages 6–10
Patna, India, December 18 - 21, 2020.

(2018) proposed an approach on NMT decoding with terminology constraints using decoder attentions which enables reduced output duplication and better constraint placement compared to existing methods. Apart from traditional approaches there is a stack-based lattice search algorithm, constraining its search space with lattices generated by phrase-based machine translation (PBMT) improves the robustness(Khayrallah et al., 2017). Wang et al. (2017) proposed two instance weighting methods with a dynamic weight learning strategy for NMT domain adaptation.

Although huge amount of research exists in this area , there exists very few works on Indian languages. As per our knowledge there is no work on technical domains like ours (Artificial Intelligence and Chemistry). Therefore there is a need to handle these technical domains and work on morphological rich and resource poor languages.

3 Approach

There are many approaches for domain adaptation discussed in section 2. However the approach we adopted , falls under combining the training data of general domain and specific technical domain data. This is further discussed in section 3.3. Our approach follows attention-based NMT implementation similar to Bahdanau et al. (2014) and Luong et al. (2015). Our model is very much similar to the model described in Luong et al. (2015) and supports label smoothing, beam-search decoding and random sampling. The brief explanation about NMT is described in section 3.1.

3.1 Neural Machine Translation

NMT system tries to find the conditional probability of target sentence with the given source sentence. In our case targets are indic languages. There are many ways to parameterize these conditional probability. Kalchbrenner and Blunsom (2013) used combination of a convolutional neural network and a recurrent neural network , Sutskever et al. (2014) used a deep Long Short-Term Memory (LSTM) model, Cho et al. (2014) used an architecture similar to the LSTM, and Bahdanau et al. (2014) used a more elaborate neural network architecture that uses an attentional mechanism over the input sequence. In this work, following Luong et al. (2015) and Sutskever et al. (2014) we used LSTM architectures for our NMT Models, which uses a LSTM to encode the input sequence and a separate LSTM to output the translation. The encoder reads the source sentence, one word at a time, and produces a large vector that represents the entire source sentence. The decoder is initialized with this vector and generates a translation, one word at a time, until it emits the end of sentence symbol. For better translations we use bi-directional LSTM (Bahdanau et al., 2014) and attention mechanism described in Luong et al. (2015).

3.2 Byte Pair Encoding (BPE)

BPE (Gage, 1994) is a data compression technique that replaces the most frequent pair of bytes in a sequence. We use this algorithm for word segmentation , and merging frequent pairs of character sequences we can get the vocabulary of desired size (Sennrich et al., 2015). As Telugu and Hindi are morphological rich languages, particularly Telugu being an Agglutinative language, therefore there is need to handle postpositions and compound words etc. BPE helps the same by separating suffix , prefix and compound words. It creates new and complex words of Telugu and Hindi language by interpreting them as sub-words units. NMT with Byte Pair Encoding made significant improvements in translation quality for low resource morphologically rich languages (Pinnis et al., 2017). We also adopted same for our experiments for all the language pairs namely English-Hindi and Hindi-Telugu. In our approach we got the best results with a vocabulary size of 20000 and dimension as 300.

3.3 Technical Domain Adaptation

Freitag and Al-Onaizan (2016) discussed two problems when we combine general data and domain data for training. First, training a neural machine translation system on large data sets can take several weeks and training a new model based on the combined training data is time consuming. Second, since the in-domain data is relatively small, the out-of-domain data will tend to dominate the training data and hence the learned model will not

perform as well on the in-domain test data.

However we preferred that approach only as our target languages are morphologically rich and resource poor languages. We addressed solutions for the above problems discussed in Freitag and Al-Onaizan (2016). First, as our main objective is to use the less amount of technical domain data(AI and Chemistry) available along with general data and improve the translation of given domain test data, adding very little amount of data will not make it more time consuming as the general data itself is less for these mentioned morphologically rich languages(Telugu and Hindi).

To address the second problem, we use BPE. Technical domain data is very very less compared to general data so if we take top 50k words as our vocabulary then most of the words will come from general data which leads to poor translation of domain data, to overcome this we used BPE as it uses sub word units and handles rare words, and it can easily recognize inflected words which are prevalent in morphologically rich languages. Due to the fact that technical domain data is very less , performing validation on combined data(general validation data + domain validation data) will lead to low translation quality for domain test data. Therefore we used only domain data for validation and got significant improvement in BLEU score on domain test data.

	Train	**Val**	**Test**
Gen-En-Hi	665474	7003	507
Gen-En-te	120708	2259	507
AI-En-Hi	4872	400	401
AI-En-te	4872	400	401
Chem-En-Hi	4984	300	397
Chem-Hi-Te	3300	300	500

Table 1: Data statistics (no. of sentences) Val-validation data Gen-general data for that language pair

4 Experiments and Results

We evaluate our approach on test data sets provided by ICON Adap-MT 2020 shared task for all language pairs for all domains. We can see data statistics in table 1. All the sentences presented in table 1 are taken from various sources provide by ICON Adap-MT 2020, these include opensubtitles, globalvoices , gnome, etc from OPUS corpus (Tiedemann, 2012). After collecting the data from above mentioned sources, training and validation data split was done based on the corpus size , then removed empty lines. To measure the translation quality we used an automatic evaluation metric called BLEU (Papineni et al., 2002).

4.1 Training Details

We have three models for each language pair 1. Baseline model trained on general data 2. Trained on general+AI data 3. general data+Chemistry data. For statistics regarding training & validation sentences refer table 1. We followed (Bahdanau et al., 2014) and (Luong et al., 2015) while training our NMT systems. Our parameters are uniformly initial- ized in [-0.1-0.1]. We used standard embedding dimension i.e 300. Comparatively we have less amount of data(including general data as well) hence we preferred to use small batch size as 10. we start with a learning rate of 0.001, for every 5 epochs we halve the learning rate. Additionally, we also use dropout with probability 0.3. In order to avoid overfitting of our models we used an early stopping criteria which is one of the forms of regularization.

Domain	**BLEU**(on val)
AI-En-Hi	8.4
Chem-En-Hi	6
AI-Hi-Te	0.6
Chem-Hi-Te	0.03

Table 2: BLEU scores of AI and Chemistry validation data on **general models** (trained on only general data) for respective language pairs

Model	**BLEU**(on val)	**BLEU**(on test)
AI-En-Hi	16	15.37
Chem-En-Hi	19.6	12.35
AI-Hi-Te	8.2	10.35
Chem-Hi-Te	5.7	6.87

Table 3: AI-En-Hi:trained on ai+gen data for English-Hindi AI-Hi-Te:trained on ai+gen data for Hindi-Telugu Chem-En-Hi:trained on chem+gen data for English-Hindi Chem-En-Hi:trained on chem+gen data for Hindi-Telugu

Source	Target	MT1	MT2
Square function is pretty simple.	स्क्वेयर फंक्शन बहुत सरल है। (skveyar phankshan bahut saral hai.)	यह काम सरल सरल है। (yah kaam saral saral hai.)	स्क्वेर फंक्शन बहुत सरल है। (skver phankshan bahut saral hai.)
In this case , there is no difference between the enzyme immunoassay and radioimmunoassay .	इस विधि में , एंजाइम इम्यूनोएसे और रेडियोइम्यूनोएसे के बीच कोई अंतर नहीं होता । (is vidhi mein , enjaim imyoonoese aur rediyoimyoonoese ke beech koee antar nahin hota.)	इस मामले में एंजाइम विज़ेशन और रेडियो के बीच कोई अंतर नहीं है। (is maamale mein enjaim vizeshan aur rediyo ke beech koee antar nahin hai.)	इस मामले में , एंजाइम इम्यूनोएसे और रेडियोइम्यूनोएसे के बीच कोई अंतर नहीं है । (is maamale mein , enjaim imyoonoese aur rediyoimyoonoese ke beech koee antar nahin hai .)

Table 4: Examples of improved sentences
MT1 : output of general model(trained on only general data)
MT2 : output of proposed model(trained on general+domain data)

4.2 Analysis

We conducted an evaluation of random sentences from the test data for both the mentioned domains, it was found that the translation of domain/technical terms or named entities was improved after adding less amount of technical domain data to the general data, we can see some of the examples in table 4 for English to Hindi for AI and Chemistry domains respectively. If we observe the first example from table 4 which is taken from AI domain, the domain term "**square function**" was translated properly into "**स्क्वेर फंक्शन**"**(skver phankshan)** when it is tested on our proposed model, same happened with chemistry domain as well, for "enzyme immunoassay" and "radioimmunoassay" domain terms, our model translated them correctly whereas the general model not. In order to show improvement in terms of bleu score, we tested our AI and Chemistry validation data on general model which was trained on only general data. Then we tested same validation data on our proposed models which trains on combining data(general+domain). When we get improvements in validation data from general model to new model, we fixed the parameters of the model as mentioned in section 3.3 for testing purpose. Table 2 shows the bleu scores of AI and Chemistry validation data on English-Hindi and Hindi-Telugu general models. Now, when we test that validation data on proposed models (table 3), the bleu score of chemistry validation data improved from **6** to **19.6** for English to Hindi language pair , in this case the bleu score increased more than three times. Similarly for AI, the bleu score increased from **8.4** to **16** for English to Hindi. For Hindi to Telugu bleu score of AI domain is increased from **0.6** to **8.2**, likewise it is increased from **0.03** to **5.7** for chemistry domain. Next we evaluated domain test data on proposed models AI-En-Hi, Chem-En-Hi, AI -Hi-Te and Chem-Hi-Te. Refer table 3 for bleu scores on test data.

5 Future Work

We would like to extend this work to possible technical domains and for more languages as well. We plan to explore many other approaches like Transformer based models for technical domain adaptation. And try to incorporate linguistic features into the NMT models.

6 Conclusion

For morphologically rich and resource poor languages like Telugu it's very difficult to get the large amount of parallel corpus for technical domain. Therefor there is a need to optimize our general models with available small amount of domain data. In this paper

we showed an approach which combines little amount of technical domain data to the available general domain data and trains a model using BPE. For better translation quality on technical domain we used only domain data as validation and observed our approach is giving promising results.

References

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

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Markus Freitag and Yaser Al-Onaizan. 2016. Fast domain adaptation for neural machine translation. *arXiv preprint arXiv:1612.06897*.

Philip Gage. 1994. A new algorithm for data compression. *C Users Journal*, 12(2):23–38.

Eva Hasler, Adrià De Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. *arXiv preprint arXiv:1805.03750*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Huda Khayrallah, Gaurav Kumar, Kevin Duh, Matt Post, and Philipp Koehn. 2017. Neural lattice search for domain adaptation in machine translation. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 20–25.

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

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.

Mārcis Pinnis, Rihards Krišlauks, Daiga Deksne, and Toms Miks. 2017. Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data. In *International Conference on Text, Speech, and Dialogue*, pages 237–245. Springer.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jorg Tiedemann. 2012. Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488.

Jiali Zeng, Yang Liu, Jinsong Su, Yubin Ge, Yaojie Lu, Yongjing Yin, and Jiebo Luo. 2019. Iterative dual domain adaptation for neural machine translation. *arXiv preprint arXiv:1912.07239*.

Domain Adaptation of NMT models for English-Hindi Machine Translation Task at AdapMT ICON 2020

Ramchandra Joshi[1], **Rushabh Karnavat**[1], **Kaustubh Jirapure**[1], **Raviraj Joshi**[2]
[1]Pune Institute of Computer Technology, Pune
[2]Indian Institute of Technology Madras, Chennai
{rbjoshi1309,rpkarnavat,kaustubhmjirapure,ravirajoshi}@gmail.com

Abstract

Recent advancements in Neural Machine Translation (NMT) models have proved to produce a state of the art results on machine translation for low resource Indian languages. This paper describes the neural machine translation systems for the English-Hindi language presented in AdapMT Shared Task ICON 2020. The shared task aims to build a translation system for Indian languages in specific domains like Artificial Intelligence (AI) and Chemistry using a small in-domain parallel corpus. We evaluated the effectiveness of two popular NMT models i.e, LSTM, and Transformer architectures for the English-Hindi machine translation task based on BLEU scores. We train these models primarily using the out of domain data and employ simple domain adaptation techniques based on the characteristics of the in-domain dataset. The fine-tuning and mixed-domain data approaches are used for domain adaptation. Our team was ranked first in the chemistry and general domain En-Hi translation task and second in the AI domain En-Hi translation task.

1 Introduction

Machine understanding of natural language queries is of paramount importance to automate different workflows. The natural language query can be in the form of text or speech. Processing of query in the form of text is more popular and easy than directly processing the raw speech waveform. The text-based Natural Language Processing (NLP) include tasks like classification, token tagging, summarization, and translation. Machine translation is an NLP technique to translate a sentence from a source language to a target language. The Neural Machine Translation (NMT) is a recent approach to translation producing state of the art results (Bahdanau et al., 2014). NMT defines translation as a sequence to sequence task and uses sequence-based neural architectures like Long Short Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997) and Transformer (Vaswani et al., 2017). Traditional techniques like rule-based translation and Statistical Machine Translation (SMT) have been outperformed by NMT models achieving significant improvements on MT tasks. In this work, we are specifically concerned with English-Hindi neural translation.

The Hindi language is one of the most popular languages in India and the fourth most spoken language in the world. Hindi is native to India and is spoken by more than 550 million total speakers worldwide. However, the number is much less as compared to global languages like English. On similar lines, the training data for the Hindi language that is publicly available for MT tasks is relatively less as compared to other highly popular languages worldwide like English, French, and German. This is important as MT tasks require a huge amount of training data to produce remarkable results using NMT models. Hindi being a relatively low resource and morphologically rich language, the amount of research in MT tasks for the Hindi language is limited (Philip et al., 2019). As Hindi is the most widely spoken language in the Indian subcontinent and the majority of content across the globe is published in English, the research in MT tasks for English-Hindi language pair becomes highly important.

Domain adaptation of translation systems to specific domains is a common practice for low resource language pairs. The adaptation is relevant as the text in different domains can vary widely (Luong and Manning, 2015). For example, social media text and the text in literary work will be quite different from style, grammar, and abbreviations perspective. The domains can be distinguished based on topics like politics, life science, news, etc, or the style of writing like formal and informal. A

Proceedings of the 17th International Conference on Natural Language Processing: Adap-MT 2020 Shared Task, pages 11–16
Patna, India, December 18 - 21, 2020.

translation model trained on one domain may not work well on other domains. The problem is more severe in models that use word-based representation as most of the domain-specific words will be out of vocabulary (Sennrich et al., 2015b). In this work, we explore ideas for domain adaptation for English-Hindi translation on the AdapMT Shared Task ICON 2020 data sets.

The AdapMT Shared Task ICON 2020 aims to evaluate the capability of general domain machine translation for Low Resource Indian Languages. Indian languages considered in AdapMT Shared Task ICON 2020 for translation are English-Hindi, English-Telugu, and Hindi-Telugu. The shared task also focuses on Low Resource domain adaptation of machine translation systems. The adaptation is done with the use of already publicly available parallel corpora and some small in-domain parallel data for AI and Chemistry domains. The creation of a publicly available parallel corpus for low resource Indian languages is another important goal of this task.

This paper describes the system built for the English-Hindi general MT and domain adaptation tasks held under AdapMT Shared Task ICON 2020. We experimented with two popular NMT models namely attention-based LSTM encoder-decoder architecture and the Transformer architecture. For domain adaption, we explore fine-tuning and mixed domain training approaches. We show that the mixed domain training performs better than the fine-tuning based approach for the datasets used in this work.

2 Architecture

In this section, we describe the two popular seq2seq neural architectures for machine translation used in this work. The encoder-decoder architecture consisting of a source side encoder and a target side decoder is used for the sequence to sequence tasks (Sutskever et al., 2014). The encoder encodes the text in a source language into a latent representation which is consumed by the decoder to generate the text in the target language. The decoder acts like a contextual language model generating target text by attending to the source representations. The attention mechanism is thus an integral part of encoder-decoder models which allows the decoder to focus on the right context while generating the corresponding target token.

2.1 LSTM model

The LSTM based encoder-decoder models use stacked LSTM layers on both encoder and decoder sides. The LSTM and GRU are commonly used recurrent neural network architectures for machine translation. In this work, we use LSTM based recurrent architecture as it is shown to give slightly better results (Britz et al., 2017). The series of stacked LSTM layers encode the source text. The hidden state of the last LSTM layer is used as the encoded output. Subsequently, the target sequence is decoded sequentially using stacked LSTM layers. The decoder also makes use of an attention mechanism to attend to the encoder's hidden state. The additive attention and dot product attention are widely used attention mechanisms (Bahdanau et al., 2014; Luong et al., 2015). In this work, we restrict ourselves to the use of additive attention.

2.2 Transformer model

The recently introduced Transformer model has found a home in almost all NLP tasks starting with neural machine translation (Vaswani et al., 2017). It has helped advance the state of the art in NLP and even employed for speech and vision tasks (Karita et al., 2019; Ramachandran et al., 2019). The Transformer uses the self-attention mechanism as the single most important component. For the task of translation, the Transformer is used on both the encoder and decoder side. It comprises various encoders and decoders stacked over each other. The main advantage of Transformer over LSTM is the parallelism on the encoder side which helps us fully exploit the underlying hardware. The multi-headed self-attention is another architectural change that helps in providing superior results as compared to LSTM.

On the encoder side, the input words are converted to vector embeddings and positional encoding is added to those embeddings so that the transformer gets the sense of the order or position of words. These embeddings are then passed on to the first encoder layer of the Transformer. The encoder consists of multi-head self-attention and a feed-forward neural network. The output from one encoder layer is given as input to the next encoder layer. The output of the final encoder layer is sent to the decoder.

The decoder consists of masked-multi head attention, multi-head Attention, and a feed-forward neural network. The embeddings along with positional

encodings are passed on to the first layer of the decoder. The masked multi-head Attention mechanism only pays attention to the previous words. Then, it is passed through the multi-head attention mechanism attending to the encoder state and a feed-forward neural network. The output of the decoder is passed to the linear and the softmax layer where the vector scores are turned into probabilities and the word with the highest probability is chosen as output.

3 Domain Adaptation

While generalization is always desirable the machine learning systems are often biased towards the domain of the training data. Each domain has a different distribution and different domain data are mixed while building general systems. In very basic terms, the vocabulary of the different domains is mostly different. Table 2 shows the percentage of in-domain tokens that are present in publicly available English-Hindi parallel training corpus. Almost 20-40% of the tokens are specific to the target domain and not present in the general corpus. Some terms are specific to and most frequently used in a particular domain. For some words, the meaning may be different across domains. For example, *"As I said that here we have one hidden layer, you can have multiple hidden layers also"* is a sentence from AI domain. Whereas, *"Triacylglycerol contains three fatty acids that are esterified to the glycerol backbone"* is from the Chemistry domain. These two sentences are very specific to their domain and rarely use in real-life conversations. To interpret them in the best way we need a domain expert or subject matter expert. Similarly to build a system that works best on a particular domain, we need to make use of domain-specific data. Now because we have the same underlying language rules irrespective of the domain we can make use of out of domain data to enhance our systems if in-domain data is less. This is exactly where domain adaptation comes into the picture.

Domain adaptation is a form of transfer learning where we adapt a general system for a specific domain. That is we tune the model to adapt to the distribution of the target domain. It has been widely studied in the context of machine translation (Chu and Wang, 2018). The adaptation techniques can either be data or model-centric. The data related approaches try to exploit the monolingual corpus of the target domain (Domhan and Hieber, 2017). A commonly used technique is to use back-translation to expand the parallel corpus of the in-domain data (Sennrich et al., 2015a). The model-based approaches also make use of monolingual corpus from the target domain to train a language model and then do a shallow or deep fusion (Gulcehre et al., 2015). There is another set of training based technique which also go into model-centric approaches. In these approaches, the model is first trained on large out of domain parallel corpus and then re-trained or fine-tuned on the small in-domain parallel corpus. There are different variations proposed in literature where the second fine-tuning is done on a mixed parallel corpus instead of only using the in-domain corpus (Chu et al., 2017). The concept of domain tag was introduced in (Sennrich et al., 2016). The model is passed the domain label along with each training sample so that it learns to distinguish between the domains. The under-represented domains are oversampled. In this work, we evaluate the domain data fine-tuning approach and mixed-data training approach. In the first approach, we train the model on general corpus followed by in-domain corpus. In the second approach, we mixed the in-domain corpus with the general data and do a single training. Since the amount of in-domain data is very less as compared to the overall general or mixed-domain data we oversample in-domain examples while training.

4 Experimental Setup

4.1 Dataset Details

In our English to Hindi machine translation experiments, we have used the publicly available IIT Bombay (IITB) English-Hindi Parallel Corpus (Kunchukuttan et al., 2017). The training data in the IITB corpus consists of nearly 1.5M training samples. The IITB training data consists of sentences from the various domain.

In addition to this, we have also used the AI and Chemistry in-domain parallel corpus provided by AdapMT Shared Task ICON 2020 organizers for training and testing the models for respective domains. The AI in-domain corpus contains 4872, 400, and 401 sentences in the train, validation, and test set, respectively. The Chemistry in-domain corpus contains 4984, 300, and 397 sentences in the train, validation, and test set, respectively. The data set details are described in Table 1.

Data	Sentences	~Tokens
IIT Bombay Train	1561840	19.85M / 21.4M
General Test	507	9k / -
AI Train	4872	77k / 83k
AI Dev	400	6k / 6k
AI Test	401	7k / -
Chemistry Train	4984	125k / 139k
Chemistry Dev	300	7k / 8k
Chemistry Test	397	7k / -

Table 1: Statistics of the Data (En / Hi)

Data	AI	Chemistry	General
Train (U)	47 / 68	64 / 60	-
Dev (U)	58 / 80	78 / 76	-
Test (U)	59 / -	55 / -	56 / -
Train	78 / 90	81 / 86	-
Dev	77 / 90	81 / 87	-
Test	78 / -	77 / -	76 / -

Table 2: Approx. % of AdapMT domain dataset tokens (En / Hi) present in IITB Train data. Rows with a suffix 'U' indicates unique tokens, while data with no suffix indicates all tokens

4.2 Data Processing

The individual data samples are lowercased followed by the removal of all the special characters. For training purposes, we exempted all the sentences from IITB English-Hindi Parallel Corpus with a length greater than 20 words. This was mainly done because of resource constraints to speed up training. After pre-processing, we train a sentence piece sub-word tokenizer to tokenize the English, as well as Hindi sentences citekudo2018sentencepiece. We train a unigram based tokenizer with a vocab size 32k (Kudo, 2018). The source and target corpus of the IITB parallel corpus was used to train the individual sentence piece models. For experiments involving domain adaptation, the domain data from the train set was also included in the sentence piece training data.

4.3 Training Details

In this paper, we used the LSTM and Transformers based models for the English to Hindi machine translation task. For the LSTM model-based experiments, we used an attention-based encoder-decoder LSTM architecture. The encoder side of LSTM is bi-directional and the decoder side of LSTM is unidirectional with Bahdanau additive attention mechanism. The number of layers on the encoder and decoder side is set to 1 with 512 hidden units in each layer. We have used a batch size of 128 and an embedding size of 256. Adam optimizer was used as an optimizer (Kingma and Ba, 2014). The subword tokenizer is used to get the subword tokens as it is known to handle the OOV problem well.

For the Transformer model, the encoder and decoder have 6 layers each and the number of hidden layers in each layer is set to 512. The batch size was set to 128. The number of heads used is 8 with a word embedding size of 512. The optimizer used was Adam. The models were implemented in Tensorflow 2.0 and trained for a maximum of 10 epochs. The validation loss was used to pick the best epoch. The standard greedy decoding was used for all the experiments. For longer sentences, during decoding, a simple heuristic to split the data at comma was used followed by separate translations. While this approach may not be well suited to the translation as the alignment is not always monotonous, it worked decently well given the nature of the in-domain sentences.

For our experiments with LSTM and Transformer models, we first trained the models on the IITB training corpus. The models are then retrained on in-domain AI and Chemistry parallel corpus to see the improvements in the machine translation model with the inclusion of small in-domain parallel data. In the second approach, the IITB corpus is mixed with the in-domain corpus individually and, a single training is performed. The in-domain corpus in oversampled 10 times to account for a very low in-domain corpus as compared to the general corpus.

5 Results and Discussion

We evaluate the mixed data and fine-tuning approaches on LSTM and Transformer NMT models. To compare the models Bilingual Evaluation Understudy (BLEU) score is used (Papineni et al., 2002). We report the BLEU score on validation data of AI and Chemistry in-domain corpus. Table 3 shows the results for de-tokenized validation data. The mixed data training approach performs the best in comparison to the no-domain data and fine-tuning approach. The no-domain data approach performs better as compared to the fine-tuning approach. This indicates that the simple fine-tuning approach is not suited to the very small in-domain corpus and is susceptible to catastrophic forgetting.

Model	AI dev	Che dev
LSTM (only IITB)	11.54	8.13
Transformer (only IITB)	10.66	4.73
LSTM (mixed)	**16.53**	**9.86**
Transformer (mixed)	12.68	5.07
LSTM (fine-tuning)	10.62	5.63
Transformer (fine-tuning)	11.60	4.88

Table 3: BLEU scores on in-domain dev data (model with the suffix 'only IITB' indicates that model is trained on samples from IITB train examples only, the model with the suffix 'mixed' indicates that the model is trained on data that is obtained by mixing oversampled in-domain training data with IITB training data, the model with suffix 'fine-tuning' indicates that the model is first trained on samples from IITB training data and then re-trained on in-domain corpus)

Data	General	AI	Chemistry
Test Data	14.81	19.08	13.95

Table 4: BLEU scores on test data as reported by AdapMT Shared Task ICON 2020 organizers

We see that although the Transformer based models perform well on the IITB test data they do not generalize well on the domain tasks. However, we feel that the low numbers with the Transformer can be enhanced using appropriate hyper-parameters and modifying the training approach. The system submitted for evaluation was LSTM based model trained on a mixed corpus which was giving the best validation scores. The results of the test system are shown in Table 4. The translations for the general test set were generated using the LSTM model trained only on the IITB parallel corpus.

6 Conclusion

In this paper, we evaluated the effectiveness of attention-based encoder-decoder LSTM and Transformer models on a low resource English to Hindi Translation Task held under AdapMT Shared Task ICON 2020. Our experiments showed that mixed domain training works well as compared to the fine-tuning approach for domain adaptation. The addition of small in-domain parallel data can indeed improve the results on AI and Chemistry domains provided in the shared task.

Acknowledgements

This work was done under the L3Cube Pune mentorship program. We would like to thank L3Cube and our mentors for the end to end guidance and encouragement to participate in the shared task.

References

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

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of simple domain adaptation methods for neural machine translation. *arXiv preprint arXiv:1701.03214*.

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

Tobias Domhan and Felix Hieber. 2017. Using target-side monolingual data for neural machine translation through multi-task learning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1500–1505.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Shigeki Karita, Nanxin Chen, Tomoki Hayashi, Takaaki Hori, Hirofumi Inaguma, Ziyan Jiang, Masao Someki, Nelson Enrique Yalta Soplin, Ryuichi Yamamoto, Xiaofei Wang, et al. 2019. A comparative study on transformer vs rnn in speech applications. In *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–456. IEEE.

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

Taku Kudo. 2018. Subword regularization: Improving neural network translation models with multiple subword candidates. *arXiv preprint arXiv:1804.10959*.

Anoop Kunchukuttan, Pratik Mehta, and Pushpak Bhattacharyya. 2017. The iit bombay english-hindi parallel corpus. *arXiv preprint arXiv:1710.02855*.

Minh-Thang Luong and Christopher D Manning. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 76–79.

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.

Jerin Philip, Vinay P Namboodiri, and CV Jawahar. 2019. A baseline neural machine translation system for indian languages. *arXiv preprint arXiv:1907.12437*.

Prajit Ramachandran, Niki Parmar, Ashish Vaswani, Irwan Bello, Anselm Levskaya, and Jonathon Shlens. 2019. Stand-alone self-attention in vision models. *arXiv preprint arXiv:1906.05909*.

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

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015b. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Controlling politeness in neural machine translation via side constraints. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 35–40.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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.

Terminology-Aware Sentence Mining for NMT Domain Adaptation: ADAPT's Submission to the Adap-MT 2020 English-to-Hindi AI Translation Shared Task

Rejwanul Haque[⋆]**, Yasmin Moslem and Andy Way**
ADAPT Centre
[⋆]School of Computing, National College of Ireland
School of Computing, Dublin City University
Dublin, Ireland
firstname.lastname@adaptcentre.ie

Abstract

This paper describes the ADAPT Centre's submission to the Adap-MT 2020 AI Translation Shared Task for English-to-Hindi. The neural machine translation (NMT) systems that we built to translate AI domain texts are state-of-the-art Transformer models. In order to improve the translation quality of our NMT systems, we made use of both in-domain and out-of-domain data for training and employed different fine-tuning techniques for adapting our NMT systems to this task, e.g. mixed fine-tuning and on-the-fly self-training. For this, we mined parallel sentence pairs and monolingual sentences from large out-of-domain data, and the mining process was facilitated through automatic extraction of terminology from the in-domain data. This paper outlines the experiments we carried out for this task and reports the performance of our NMT systems on the evaluation test set.

1 Introduction

ADAPT Centre participated in the Adap-MT 2020 Translation Shared Task[1] of the 17th International Conference on Natural Language Processing (ICON 2020).[2] This task aims at evaluating machine translation (MT) systems on the translation of documents from two domains (AI and Chemistry) involving low-resource Indic languages. The task addresses a number of translation directions, and we participated in the English-to-Hindi translation task and focused on translating the AI texts only. To make the readers familiar with the AI translation task and to understand the challenges of this task, we show a couple of sentences from the blind test set in Table 1.

(1) Machine learning (ML) is a branch of AI that allows chatbots to identify patterns in human language and learn from past conversations.
(2) Approaches include statistical methods, computational intelligence, and traditional symbolic AI.

Table 1: Sentences from the AI blind test set.

Our MT systems are Transformer models (Vaswani et al., 2017) which were trained using the Marian-NMT toolkit.[3] In this work, we applied different data augmentation and domain adaptation techniques to train our models, such as using synthetic data from target-side monolingual data through the use of back-translation (Sennrich et al., 2016a; Poncelas et al., 2018), mixed fine-tuning (Chu et al., 2017) and on-the-fly model adaption (Chinea-Ríos et al., 2017). As for the latter two approaches, we mined sentences and sentence pairs from large out-of-domain monolingual and parallel corpora, respectively, based on domain terms appearing in the in-domain data. Note that the terms were extracted automatically from the in-domain data.

This remainder of the paper is organized as follows. Section 2 presents our approaches. We describe the resources we utilized for training in Section 3. Section 4 presents the results obtained, and Section 5 concludes our work with avenues for future work.

2 Our Approaches

2.1 Training Data Augmentation

The use of unlabeled monolingual data in addition to limited bitexts for NMT training (Sennrich et al.,

[1] https://ssmt.iiit.ac.in/machinetranslation.html
[2] https://www.iitp.ac.in/~ai-nlp-ml/icon2020/main_prog.html
[3] https://github.com/marian-nmt/marian

Proceedings of the 17th International Conference on Natural Language Processing: Adap-MT 2020 Shared Task, pages 17–23
Patna, India, December 18 - 21, 2020.

2016a; Zhang and Zong, 2016; Burlot and Yvon, 2018; Poncelas et al., 2018; Caswell et al., 2019) is nowadays a common practice in MT development (Barrault et al., 2020). This has even more impact when applied to the specialised domains and many language pairs, for which obtaining parallel data is a challenge.

In this task, in order to improve our baseline English-to-Hindi Transformer model, we augmented our training data with target-original synthetic data. As in Caswell et al. (2019), in order to let the NMT model know that the given source is synthetic, we tag the source sentences of the synthetic data with the extra tokens. Iterative generation and training on synthetic data can yield increasingly better NMT systems, especially in low-resource scenarios (Hoang et al., 2018; Chen et al., 2019). Since our baseline target-to-source (Hindi-to-English) MT system is already good in quality, it was used to translate the Hindi monolingual data.

2.2 Mixed Fine-Tuning

As for adapting our baseline MT model to the AI domain, we implemented mixed fine-tuning of model parameters, where fine-tuning is conducted on the training data that consists of both in-domain and out-of-domain data as described in Chu et al. (2017). The shared task organisers released parallel training data of the AI domain with a limited number of in-domain examples (only 4,872 sentence pairs). The in-domain data was augmented by oversampling the AI training set several times, and an almost similar sized out-of-domain data set is mined from the parallel (out-of-domain) training corpus on which our baseline NMT system was trained. This strategy worked well for us when we translated business scene dialogue (Jooste et al., 2020) in the WAT 2020[4] (Nakazawa et al., 2020) document-level translation task. However, the adaptation method presented in this paper slightly differs from the conventional mixed fine-tuning (Chu et al., 2017; Jooste et al., 2020), and is described below.

Terms are usually indicators of the nature of a domain and play a critical role in domain-specific MT (Haque et al., 2019, 2020a). Sentences that contain in-domain terms are likely to be in-domain sentences. However, an ambiguous term could have more than one potential meaning. As an example of lexical ambiguity, 'cold' has several possible meanings in the Unified Medical Language System Metathesaurus (Humphreys et al., 1998) including 'common cold', 'cold sensation' and 'cold temperature' (Stevenson and Guo, 2010). Moreover, a polysemous term (e.g. 'cold') could have many translation equivalents in a target language. With this in mind, we mined those training examples (i.e. sentence pairs) from the large out-of-domain domain parallel corpus whose source or target sentences contain at least one domain term. As pointed out earlier, an extracted out-of-domain sentence that contain a domain term may not represent the desired domain; however, the training examples that include such sentences may play a crucial role in minimising lexical selection errors as far as terminology translation in NMT is concerned (Haque et al., 2019, 2020a).

To this end, we exploit the approaches of Rayson and Garside (2000) and Haque et al. (2014, 2018) in order to automatically identify terms in the in-domain texts. The idea is to identify those words which are most indicative (or characteristic) of the in-domain corpus compared to a reference corpus. Haque et al. (2014, 2018) used a large corpus which is generic in nature as a reference corpus. We adopted their approach and used a large generic corpus in order to identify terms in the in-domain source (English) and target (Hindi) corpora. In our setup, we also used the source and target sides of the out-of-domain training bitexts on which our baseline NMT system was trained as the reference corpora. The intuition is again the same, i.e. to extract those (terminological) expressions from the in-domain data that do not occur or rarely occur in the training data and are more indicative of the in-domain AI corpus. Given the lists of source and target terms, we mine sentences independently from the source and target sides of the out-of-domain bilingual corpus. As pointed out above, we select those sentence pairs from the out-of-domain bilingual corpus whose source or target sides contain at least one domain term. In Nayak et al. (2020b), we empirically showed that such "pseudo" in-domain sentences are more effective than those mined using bilingual cross-entropy difference according to the in-domain language model (Axelrod et al., 2011) for NMT model adaptation.

As in Kobus et al. (2017), in order to inform the NMT model about the domain during training and decoding, we add a (domain) tag at the begin-

[4] http://lotus.kuee.kyoto-u.ac.jp/WAT/WAT2020/index.html

ning of the source sentences of the in-domain data, which allows us to control the output domain of the trained system. The NMT system is finally fine-tuned on the mixture of the in-domain and mined out-of-domain corpora.

2.3 Mining Sentences for Fine-tuning

Chinea-Ríos et al. (2017) demonstrated that in the case of specialised domains where parallel corpora are scarce, sentences of a large monolingual data that are more related to the test set sentences to be translated could be effective for fine-tuning the original general domain NMT model. They select those instances from a large monolingual corpus whose vector-space representation is similar to the representation of the test set instances. The selected sentences are then automatically translated by an NMT system built on a general domain data. Finally, the NMT system is fine-tuned with the resultant synthetic data. The synthetic training data whose source-side sentences are original could be more effective for domain adaptation, and the learning method that uses such training data is called 'self-training' (Ueffing et al., 2007). In a similar line of research, it has also been shown that an NMT system built on general domain data can be fine-tuned using just a few sentences (Farajian et al., 2017; Wuebker et al., 2018; Huck et al., 2019).

We followed Chinea-Ríos et al. (2017) in order to mine those sentences from large monolingual datasets that could be beneficial for fine-tuning the original NMT model. As in Jooste et al. (2020); Nayak et al. (2020b); Parthasarathy et al. (2020), we first identified terms in the AI test set to be translated, and given the list of extracted terms, English sentences which were mined from large monolingual data are similar in style to the AI test set sentences. To put it another way, we followed the method described in Section 2.2 in order to extract sentences form large monolingual corpus. The monolingual corpus that we used for this purpose contains 95,918,840 sentences which were sampled from CommonCrawl[5] and Wikipedia Dumps.[6] The English source sentences that have been mined were translated into Hindi using the best MT system (cf. through mixed fine-tuning strategy) to create synthetic data (i.e. source-side original synthetic corpus (SOSC)) to be used for fine-tuning the same NMT model.

3 Data Used and Training Setups

For building our baseline models (forward and backward), we used only the bilingual data provided by the task organisers. As for Hindi monolingual sentences for back-translation, we sampled them from AI4Bharat-IndicNLP Corpus (Kunchukuttan et al., 2020). The out-of-domain parallel data is compiled from a variety of existing sources, e.g. OPUS[7] (Tiedemann, 2012), and after applying standard cleaning procedures including applying a language identifier[8] we are left with just over 1.1 million parallel sentence pairs. Table 2 presents the corpus statistics. The development set

In-domain	sentences	words (EN)	words (HI)
Train	4,872	77,301	82,815
Development	400	7,031	7,064
Out-of-domain	1,102,511	22.4M	23.4M
Hindi Monolingual			
Setup 1	1M		18.8M
Setup 2	7.82M		142.9M

Table 2: The Corpus statistics.

(cf. Table 2) of the AI English-to-Hindi translation task consists only of 400 sentence pairs. For experimentation, we considered its first 200 sentence pairs as development set and the remainder as the evaluation test set. We used two different sized monolingual datasets for our back-translation experiments (cf. last rows of Table 2).

As pointed out earlier, our NMT systems are Transformer models. The tokens of the training, evaluation and validation sets are segmented into sub-word units using Byte-Pair Encoding (BPE) (Sennrich et al., 2016b), and BPE is applied individually on the source and target languages. From our experiences (Jooste et al., 2020; Haque et al., 2020b; Nayak et al., 2020b,a; Parthasarathy et al., 2020) in the participation in the recent shared translation tasks (Barrault et al., 2020; Mayhew et al., 2020; Nakazawa et al., 2020) involving low-resource language pairs and domains, we found that the following configuration usually leads to the best results in our low-resource translation settings: (i) the BPE vocabulary size: 6,000, (ii) the sizes of the encoder and decoder layers: 4 and 6,

[5] http://web-language-models.s3-website-us-east-1.amazonaws.com/wmt16/deduped/en-new.xz

[6] http://data.statmt.org/wmt20/translation-task/ps-km/wikipedia.en.lid_filtered.test_filtered.xz

[7] http://opus.lingfil.uu.se/

[8] https://pypi.org/project/pycld2/

respectively, and (iii) learning-rate: 0.0003. As for the remaining hyperparameters, we followed the recommended best setup from Vaswani et al. (2017). The early stopping criterion is based on cross-entropy; however, the final NMT system is selected as per the highest BLEU score on the validation set. The beam size for search is set to 6. We make our final NMT model with ensembles of 8 models that are sampled from the training run.

4 Experiments and Results

This section presents the performance of our MT systems in terms of the automatic evaluation metric BLEU (Papineni et al., 2002). Additionally, we performed statistical significance tests using bootstrap resampling methods (Koehn, 2004). We obtained the BLEU scores of our MT systems to evaluate them on the test set, and the scores are reported in Table 3. The first row of Table 3 represents our baseline English-to-Hindi MT system. The Hindi-to-English MT system which has been used to translate the Hindi monolingual sentences to English is of good quality (i.e. it produces 28.76 BLEU points on the test set). The BLEU scores of the MT systems (Base2 and Base3) trained on training data that consists of both authentic and synthetic parallel data are shown in the next two rows of Table 3 (cf. Section 2.1).

	BLEU
Base	28.97
Base2 (Base + 1M Syn)	30.80
Base3 (Base + 8M Syn)	29.97
Base2 + Mixed FT	42.02
Base3 + Mixed FT	43.03
Base2 + Mixed FT + ST	43.00
Base3 + Mixed FT + ST	43.51

Table 3: The BLEU scores of the English-to-Hindi NMT systems.

Source–target sentence pairs were mined from out-of-domain training bitexts for mixed fine-tuning (see Section 2.2). The number of sentence pairs that have been mined is 167,234. We also augmented the in-domain parallel corpus via over-sampling in-domain sentences, and by this, the size of the in-domain bitexts becomes 97,440. We finally fine-tuned Base2 and Base3 on the training data that is a mixture of (augmented) in-domain and (mined) out-of-domain data. The BLEU scores of the MT systems (Base2 + Mixed FT and Base3 + Mixed FT) which are the results of the fine-tuning process are presented in the fourth and fifth rows of Table 3. One of our three submission (*Run1*) is with Base3 + Mixed FT. We select Base2 + Mixed FT and Base3 + Mixed FT for further adaptation.

Following the method described in Section 2.3, we mined English sentences (a total of 27,644 sentences) from a large monolingual corpus (cf. Section 2.3) given the list of terms (a total of 356 terms) appearing in the test set. Then, SOSC was created by translating these mined English sentences into Hindi using the respective MT system. Finally, the best MT systems (Base2 + Mixed FT or Base3 + Mixed FT) were fine-tuned on the resultant SOSC. The BLEU scores of the adapted MT systems on the test set are shown in the last rows of Table 3. When we compare the original MT systems with the adapted MT systems, we see that (i) the adapted version of Base2 + Mixed FT, Base2 + Mixed FT + ST, produces a 0.98 BLEU point (corresponding to 2.33% relative) improvement over Base2 + Mixed FT, and (ii) the same of Base3 + Mixed FT, Base3 + Mixed FT + ST, produces a 0.48 BLEU point (corresponding to 1.1% relative) improvement over Base3 + Mixed FT. The former improvement is statistically significant but the latter is not.

As above, we created the adapted MT systems for the blind test set which consists of 401 sentences. Our terminology extraction model identified 1,599 AI terms in the blind test set. We mined 98,009 English sentences from the large monolingual data given the list of terms. We followed the approach described above for fine-tuning our best two models (Base2 + Mixed FT and Base3 + Mixed FT) in order to translate the blind test set sentences. The BLEU scores of our MT systems on the blind test set, which the task organisers published, are shown in Table 4.

MT systems	Submissions	BLEU
Base2 + Mixed FT	*Run1*	35.78
Base2 + Mixed FT + ST	*Run2*	36.71
Base3 + Mixed FT + ST	*Run3*	39.15

Table 4: The BLEU scores of the MT systems on the blind test set.

5 Conclusion

In this paper, we described our MT systems that were submitted to the Adap-MT 2020 AI translation shared task. We presented our results obtained

at the time of development of our MT systems. In order to adapt our MT systems to translate texts of AI domains, we subsequently applied two existing fine-tuning techniques while using a term extraction model in the translation pipeline for mining sentences similar to the domain and style of those of the AI data. We showed that, in the case of limited in-domain training data, both out-of-domain data which are selected via term-based mining protocol and in-domain data are useful for fine-tuning model parameters, which essentially provides our best results in this translation task. Furthermore, making use of synthetic parallel data in training also greatly increased the performance of our MT systems. As for the shared task's system rankings, our three submissions *Run3*, *Run2* and *Run1* secured second, third and fourth positions, respectively.

In future, we aim to apply our strategy to other domains and language pairs.

Acknowledgments

The ADAPT Centre for Digital Content Technology is funded under the Science Foundation Ireland (SFI) Research Centres Programme (Grant No. 13/RC/2106) and is co-funded under the European Regional Development Fund. The publication has emanated from research supported in part by a research grant from SFI under Grant Number 13/RC/2077 and 18/CRT/6224.

References

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

Loïc Barrault, Magdalena Biesialska, Ondřej Bojar, Marta R. Costa-jussà, Christian Federmann, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Matthias Huck, Eric Joanis, Tom Kocmi, Philipp Koehn, Chi-kiu Lo, Nikola Ljubešić, Christof Monz, Makoto Morishita, Masaaki Nagata, Toshiaki Nakazawa, Santanu Pal, Matt Post, and Marcos Zampieri. 2020. Findings of the 2020 Conference on Machine Translation (WMT20). In *Proceedings of the Fifth Conference on Machine Translation*, pages 1–54, Online. Association for Computational Linguistics.

Franck Burlot and François Yvon. 2018. Using Monolingual Data in Neural Machine Translation: a Systematic Study. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 144–155, Belgium, Brussels. Association for Computational Linguistics.

Isaac Caswell, Ciprian Chelba, and David Grangier. 2019. Tagged Back-Translation. In *Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers)*, pages 53–63, Florence, Italy. Association for Computational Linguistics.

Peng-Jen Chen, Jiajun Shen, Matthew Le, Vishrav Chaudhary, Ahmed El-Kishky, Guillaume Wenzek, Myle Ott, and Marc'Aurelio Ranzato. 2019. Facebook AI's WAT19 Myanmar-English Translation Task Submission. In *Proceedings of the 6th Workshop on Asian Translation*, pages 112–122, Hong Kong, China. Association for Computational Linguistics.

Mara Chinea-Ríos, Álvaro Peris, and Francisco Casacuberta. 2017. Adapting Neural Machine Translation with Parallel Synthetic Data. In *Proceedings of the Second Conference on Machine Translation*, pages 138–147, Copenhagen, Denmark. 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 (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-Domain Neural Machine Translation through Unsupervised Adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137, Copenhagen, Denmark. Association for Computational Linguistics.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2019. Investigating Terminology Translation in Statistical and Neural Machine Translation: A Case Study on English-to-Hindi and Hindi-to-English. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2019)*, pages 437–446, Varna, Bulgaria. INCOMA Ltd.

Rejwanul Haque, Mohammed Hasanuzzaman, and Andy Way. 2020a. Analysing Terminology Translation Errors in Statistical and Neural Machine Translation. *Machine Translation (in press)*, 34.

Rejwanul Haque, Yasmin Moslem, and Andy Way. 2020b. The ADAPT System Description for the STAPLE 2020 English-to-Portuguese Translation Task. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 144–152, Online. Association for Computational Linguistics.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2014. Bilingual Termbank Creation via Log-Likelihood

Comparison and Phrase-Based Statistical Machine Translation. In *Proceedings of the 4th International Workshop on Computational Terminology (Computerm)*, pages 42–51, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Rejwanul Haque, Sergio Penkale, and Andy Way. 2018. TermFinder: log-likelihood comparison and phrase-based statistical machine translation models for bilingual terminology extraction. *Language Resources and Evaluation*, 52(2):365–400.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative Back-Translation for Neural Machine Translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Matthias Huck, Viktor Hangya, and Alexander Fraser. 2019. Better OOV Translation with Bilingual Terminology Mining. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5809–5815, Florence, Italy. Association for Computational Linguistics.

Betsy L. Humphreys, Donald A. B. Lindberg, Harold M. Schoolman, and G. Octo Barnett. 1998. The Unified Medical Language System: An Informatics Research Collaboration. *Journal of the American Medical Informatics Association*, 5(1):1–11.

Wandri Jooste, Rejwanul Haque, and Andy Way. 2020. The ADAPT Centre's Neural MT Systems for the WAT 2020 Document-Level Translation Task. In *Proceedings of the the 7th Workshop on Asian Translation (WAT 2020), AACL-IJCNLP 2020*, page (in press), Suzhou, China.

Catherine Kobus, Josep Crego, and Jean Senellart. 2017. Domain Control for Neural Machine Translation. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 372–378, Varna, Bulgaria. INCOMA Ltd.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Anoop Kunchukuttan, Divyanshu Kakwani, Satish Golla, Avik Bhattacharyya, Mitesh M Khapra, Pratyush Kumar, et al. 2020. AI4Bharat-IndicNLP Corpus: Monolingual Corpora and Word Embeddings for Indic Languages. *arXiv preprint arXiv:2005.00085*.

Stephen Mayhew, Klinton Bicknell, Chris Brust, Bill McDowell, Will Monroe, and Burr Settles. 2020. Simultaneous Translation and Paraphrase for Language Education. In *Proceedings of the Fourth Workshop on Neural Generation and Translation*, pages 232–243, Online. Association for Computational Linguistics.

Toshiaki Nakazawa, Hideki Nakayama, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Anoop Kunchukuttan, Shantipriya Parida, Ondřej Bojar, and Sadao Kurohashi. 2020. Overview of the 7th Workshop on Asian Translation. In *Proceedings of the 7th Workshop on Asian Translation*, Suzhou, China. Association for Computational Linguistics.

Prashanth Nayak, Rejwanul Haque, and Andy Way. 2020a. The ADAPT Centre's Participation in WAT 2020 English-to-Odia Translation Task. In *Proceedings of the the 7th Workshop on Asian Translation (WAT 2020), AACL-IJCNLP 2020*, page (in press), Suzhou, China.

Prashanth Nayak, Rejwanul Haque, and Andy Way. 2020b. The ADAPT's submissions to the WMT20 biomedical translation task. In *Proceedings of the Fifth Conference on Machine Translation (Shared Task Papers (Biomedical)*, Punta Cana, Dominican Republic.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Venkatesh Balavadhani Parthasarathy, Akshai Ramesh, Rejwanul Haque, and Andy Way. 2020. The ADAPT system description for the WMT20 news translation task. In *Proceedings of the Fifth Conference on Machine Translation (Shared Task Papers (News))*, Punta Cana, Dominican Republic.

Alberto Poncelas, Dimitar Shterionov, Andy Way, Gideon Maillette de Buy Wenniger, and Peyman Passban. 2018. Investigating Backtranslation in Neural Machine Translation. In *Proceedings of The 21st Annual Conference of the European Association for Machine Translation (EAMT 2018)*, pages 249–258, Alicante, Spain.

Paul Rayson and Roger Garside. 2000. Comparing Corpora using Frequency Profiling. In *The Workshop on Comparing Corpora*, pages 1–6, Hong Kong, China. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. 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, Berlin, Germany. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Mark Stevenson and Yikun Guo. 2010. Disambiguation of ambiguous biomedical terms using examples generated from the UMLS Metathesaurus. *Journal of Biomedical Informatics*, 43(5):762–773.

Jörg Tiedemann. 2012. Parallel Data, Tools and Interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC-2012)*, pages 2214–2218, Istanbul, Turkey. European Languages Resources Association (ELRA).

Nicola Ueffing, Gholamreza Haffari, and Anoop Sarkar. 2007. Transductive learning for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 25–32, Prague, Czech Republic. Association for Computational Linguistics.

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 6000–6010.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting Source-side Monolingual Data in Neural Machine Translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

MUCS@Adap-MT 2020: Low Resource Domain Adaptation for Indic Machine Translation

Asha Hegde
Department of Computer Science
Mangalore University
hegdekasha@gmail.com

H. L. Shashirekha
Department of Computer Science
Mangalore University
hlsrekha@gmail.com

Abstract

Machine Translation (MT) is the task of automatically converting the text in source language to text in target language by preserving the meaning. MT task usually require large corpus for training the translation models. Due to scarcity of resources very less attention is given to translating into low resource languages and in particular into Indic languages. In this direction, a shared task called "Adap-MT 2020: Low Resource Domain Adaptation for Indic Machine Translation" is organized to illustrate the capability of general domain MT when translating into Indic languages and low resource domain adaptation of MT systems. In this paper, we, team MUCS, describe a simple word extraction based domain adaptation approach applied to English-Hindi MT only. MT in the proposed model is carried out using Open-NMT - a popular Neural Machine Translation tool. A general domain corpus is built effectively combining the available English-Hindi corpora and removing the duplicate sentences. Further, domain specific corpora is updated by extracting the sentences from generic corpus that match with the vocabulary of the domain specific corpus. The proposed model is exhibited satisfactory results for small domain specific AI and CHE corpora in terms of Bilingual Evaluation Understudy (BLEU) score with 1.25 and 2.72 respectively. Further, this methodology is quite generic and can easily be extended to other low resource language pairs as well.

1 Introduction

Machine Translation (MT) acts as a bridge for cross-language communication in Natural Language Processing (NLP). It handles perplexity problems between two languages while preserving its meaning. MT was one of the initial tasks taken up by computer scientists and the research in this field is going on for last 50 years. MT task was initially handled with dictionary matching techniques and slowly upgraded to rule-based approaches (Dove et al., 2012). To resolve knowledge acquisition issues corpus based approaches became popular and bilingual parallel corpora was used to acquire translation knowledge (Britz et al., 2017). Along with corpus based approaches, hybrid MT approaches also became popular as these approaches promise state-of-the-art result.

The recent shift to large-scale analytical techniques has resulted in very significant improvements in the quality of MT. Neural Machine Translation (NMT) - a corpus based approach has gained attention of the MT researchers. NMT is the task of translating text from one natural language (source) to another natural language (target) using most commonly, Recurrent Neural Networks (RNN), specifically the Encoder-Decoder or Sequence-to Sequence models (Sutskever et al., 2014). Further, unlike conventional translation systems, all parts of the neural translation model are trained jointly (end-to-end) to maximize the translation performance (Bahdanau et al., 2014). In an NMT system, a bidirectional RNN, known as encoder is used by the Neural Network (NN) to encode a source sentence for a second RNN, known as decoder which is used to predict words in the target language. This encoder-decoder architecture can be designed with multiple layers to increase the efficiency of the system. Now, NMT has become an effective alternative to traditional Phrase-Based Statistical Machine Translation (Patil and Davies, 2014).

1.1 Challenges of NMT

In spite of its popularity, NMT faces the following challenges

- Normally NMT require a large dataset for training the model and powerful computa-

Proceedings of the 17th International Conference on Natural Language Processing: Adap-MT 2020 Shared Task, pages 24–28
Patna, India, December 18 - 21, 2020.

tional resource to build NN with sufficient amount of hidden layers.

- NMT is inconsistent in handling rare words. Since these words are sparsely available in the network, learning and inferencing them is not efficient.

- Though many experiments are being carried out to handle long sentences, long term dependency issue is still considered as a major problem in NMT (Tien and Minh, 2019).

The main objective of this work is to investigate efficient strategies to perform English to Hindi MT using sufficient amount of general domain corpora and very small domain specific corpora. Rest of the paper is structured as follows: Section 2 gives the brief description about domain adaptation and different approaches to domain adaptation followed by the methodology in Section 3. Experiments and results are given in Section 4 and conclusion in Section 5.

2 Domain Adaptation for NMT

Dataset plays a crucial role in NN based translation models. Huge amount of quality dataset for training results in good translation performance whereas small dataset results in poor translation performance. Hence, if the dataset is small, effective management of such dataset for NN based translation will be the key for better translation performance. Domain adaptation techniques that transfer existing knowledge to new domains as much as possible is one method in this direction. Domain Adaptation (DA) is a sub-discipline of machine learning in which a model trained on a source distribution is used in the context of a different (but related) target distribution. In simple words, it is the ability to apply an algorithm trained in one domain to a different domain or updating one corpus using another corpus.

While the big generic corpus will help to avoid out-of-vocabulary problem and unidiomatic translations, the small specialized corpus will help to capture terminology and vocabulary that is required for the translation (Šoštarić et al., 2019). Few effective DA approaches which promise better translation performance are as follows:

- Incremental Training and Re-training - In this approach, initially a model is trained on a huge generic corpus and then the same model is re-trained on a small domain specific corpus. This approach has two phases: i) pre-processing and training of huge generic corpus and ii) pre-processing the new domain specific corpus and re-training the base model on the domain specific corpus (Kalimuthu et al., 2019).

- Ensemble of decoding - In this approach, the base model is trained on generic dataset and the model is re-trained on domain specific dataset. Then instead of combining dataset, both the models are combined during translation (Chu and Wang, 2018).

- Combining Training Data - This approach is a simple and effective DA approach compared to all other approaches. In this approach, both the corpora are combined and this new corpus is used for training ie., huge generic corpus is combined with domain specific corpus and then this new corpus is used for training (Chu and Wang, 2018).

- Data Augmentation - In this approach, size of the domain specific dataset is increased using phrase based translation technique. The information related to word alignment is extracted from the corpus and then this information is used to build n-gram model to construct new dataset. Further, duplicates are discarded to avoid redundancy (Xia et al., 2019).

Table 1: Details of General domain English-Hindi parallel corpus

Resource	**No. of parallel sentences**	**No. of words**
IIT Bombay	2,00,000	6,28,56,567
Bible	62,073	4,10,589
globalvoices	2,299	1,70,116
CVIT-MKB	5,272	3,54,128

3 Methodology

Despite the considerable advances in MT models, translation of low-resource languages is still an unresolved issue and DA approaches are promising considerable performance in this direction. In the proposed work, a DA approach of combining both generic dataset and domain specific dataset based on the vocabulary of domain specific dataset is

used to conduct effective training and inference for translation using openNMT- a popular open source tool (Klein et al., 2018). OpenNMT accepts only primarily cleaned dataset as its input. Therefore, noise such as initial space, end space, blank lines and special characters have been removed from the bilingual parallel corpus. This pre-processing is carried out for both generic corpus and domain specific corpora. Then vocabulary of the domain specific corpora is constructed and sentences that contain any of the words in this vocabulary are extracted from the generic corpus. Finally, these extracted sentences are added to the domain specific corpus and the updated corpus is used to train the translation model. Table 2 illustrates a sample sentence from generic corpus and from domain specific corpus along with their vocabulary. The word 'queen' which is present in domain specific corpus is also present in the generic corpus. Hence, that sentence from the generic corpus will be extracted and added to the domain specific corpus.

3.1 Dataset

Dataset and the preparation of dataset for training the translation model play a major role in MT. This data preparation process is carried out at different levels to conduct effective translation.

General domain English-Hindi corpus is constructed by combining various open source corpora namely English-Hindi parallel corpus open sourced by IIT Bombay[1], English-Hindi bible corpus[2], Globalvoices[3] and CVIT-MKB[4]. Then this newly constructed generic corpus is pre-processed so that the corpus can be used to train in openNMT. Sufficient training and validation dataset is used which is the basic requirement of openNMT.

AI English-Hindi corpus is pre-processed and combined with general domain English-Hindi corpus based on the vocabulary of AI English-Hindi corpus. Then this new corpus is used for translation in openNMT model.

Chemistry English-Hindi corpus is pre-processed and combined with general domain English-Hindi corpus based on the vocabulary of CHE English-Hindi corpus. Then this new corpus is used for translation in openNMT model.

Details of general domain English-Hindi parallel corpus are shown Table 1 and details of AI and CHE corpora are shown in Table 4. Table 3 shows the details of domain specific dataset after applying DA and details of train and validation dataset used for the experiments are shown in Table 6.

4 Experimental setup

English to Hindi MT is implemented using openNMT which is considered as the most sophisticated generalized translation tool that provides easy modifications. As this model requires GPU, we set up this experiment in Google colaboratory. Translation experiments are carried out by continuous tuning of the model to conduct better training. Initially, this model is trained using a huge generic corpus then the same set up is used for domain specific corpus. As the given domain specific corpora are very small to conduct efficient translation, training data of domain specific corpora is combined with generic corpus based on vocabulary of the domain specific corpora and the training is continued with the same set up.

4.1 Result

The proposed model predicts Hindi sentences for the given English test sentences and the sample snapshot of the model is shown in Figure 1 and the performance measure of the proposed model in terms of accuracy and perplexity is shown in Table 5. Further, the proposed system is evaluated separately using BLEU score (Papineni et al., 2002) for both generic corpus and domain specific corpora. Though there are many challenges with the test dataset, considerable results are obtained for both generic corpus and domain specific corpora.

4.2 Result Analysis

The results obtained for the given test set with respect to general domain corpus shows 63.43% accuracy with 20.51 perplexity using openNMT model. This model shows considerable accuracy for the generic corpus as it contains lots of challenges related to alignment, mixing of different script, length of the sentences etc. Then, the results obtained for translating the given test set with respect to domain specific AI corpus in the same setup shows 30.63% accuracy with 45.68 perplexity. As this corpus is very small to conduct translation the same is replicated in the result ie., it exhibits poor translation. Then, after applying proposed DA approach the model shows improvement in both accuracy

[1] http://www.cfilt.iitb.ac.in/iitb$_parallel$
[2] http://opus.nlpl.eu/bible-uedin.php
[3] http://opus.nlpl.eu/GlobalVoices.php
[4] http://preon.iiit.ac.in/ jerin/bhasha/

Table 2: Sample sentences

corpus	Sentence	Vocabulary
Generic corpus	The **Queen** said: Know my nobles that a gracious letter has been delivered to me.	**queen**, said, know, nobles, gracious, letter, delivered
Domain specific corpus	Example one, in a bee hive, there are many thousands of workers bee that all serve one **queen** bee	example, one, bee, hive, thousands, workers, serve, **queen**

SENT 16: ['The', 'interactions,', 'reactions', 'and', 'transformations', 'that', 'are', 'studied',

'in', 'chemistry', 'are', 'usually', 'the', 'result', 'of', 'interactions', 'between', 'atoms,', 'leading',

'to', 'rearrangements', 'of', 'the', 'chemical', 'bonds', 'which', 'hold', 'atoms', 'together.']

PRED 16: स्टेबिलिटी और तहपत्हेस के बीच में जो कार्बोहाइड्रेट्स का अध्ययन करते हैं, वे हाइड्रोजन बॉन्ड्स के बीच होते हैं, जो हाइड्रोजन बॉन्ड्स के बीच हाइड्रोजन बॉन्ड्स होते हैं।

PRED SCORE: -45.7755

SENT 17: ['Such', 'behaviors', 'are', 'studied', 'in', 'a', 'chemistry', 'laboratory.']

PRED 17: यह chemistry laboratory. में लिखा हुआ लिखा है।

PRED SCORE: -10.0361

SENT 18: ['The', 'chemistry', 'laboratory', 'stereotypically', 'uses', 'various', 'forms', 'of',

'laboratory', 'glassware.']

PRED 18: जेवनारों में गवैयों का निर्माण करने के लिए पृथक्करण का निर्माण किया जाता है।

PRED SCORE: -26.7975

Figure 1: Predicted English-Hindi sentences using openNMT

Table 3: Details of domain specific English-Hindi parallel corpora after domain adaptation (for training)

Corpus name	No. of parallel sentences	No. of words	Vocab Size
AI	2,28,079	6,66,42,961	98,606
CHE	2,27,873	6,62,58,875	1,00,006

Table 4: Details of domain specific English-Hindi parallel corpora before domain adaptation (for training)

Corpus name	No. of parallel sentences	No. of words
AI	4,383	8,05,483
CHE	3,567	13,72,980

and perplexity ie., 41.98% and 38.52 respectively. Because of DA technique used for translation, domain specific dataset is increased to capture rare words that improves the translation. Further, the results obtained for translating the given test set with respect to domain specific CHE corpus using

Table 5: Performance measurement of the model

Corpus Name	Accuracy	Perplexity
Generic Corpus	63.43	20.51
AI (Before DA)	30.63	45.68
CHE (Before DA)	31.57	40.48
AI (After DA)	41.98	38.52
CHE (After DA)	42.87	29.25

Table 6: Details of training and validation sentences used for the model

Corpus name	**No. of Training sentences**	**No. of validation sentences**
Generic	2,69,400	20,244
AI	2,65,383	20,400
CHE	2,46,867	20,300

openNMT shows 31.57% accuracy with 40.48 perplexity. Then, proposed DA approach is applied and newly constructed corpus is used in the model. It shows improvement in both accuracy and perplexity ie., 42.87% and 29.25 respectively.

5 Conclusion and Future work

In this English-Hindi translation work, a huge generic corpus and small domain specific corpora are used for translation in openNMT. Further, a simple domain adaptation technique is used to tackle translation issues of low-resource languages. As this approach is language independent it can easily be extended to other low-resource languages. Further, these experiments have exhibited satisfactory results for both generic corpus and domain specific corpora.

We would like to explore different preprocessing techniques that helps to translate low resource languages efficiently.

References

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

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. *arXiv preprint arXiv:1703.03906*.

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

Catherine Dove, Olga Loskutova, and Ruben de la Fuente. 2012. What's your pick: Rbmt, smt or hybrid. In *Proceedings of the tenth conference of the Association for Machine Translation in the Americas (AMTA 2012). San Diego, CA*.

Marimuthu Kalimuthu, Michael Barz, and Daniel Sonntag. 2019. Incremental domain adaptation for neural machine translation in low-resource settings. In *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pages 1–10.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander M Rush. 2018. Opennmt: Neural machine translation toolkit. *arXiv preprint arXiv:1805.11462*.

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.

Sumant Patil and Patrick Davies. 2014. Use of google translate in medical communication: evaluation of accuracy. *Bmj*, 349:g7392.

Margita Šoštarić, Nataša Pavlović, and Filip Boltužić. 2019. Domain adaptation for machine translation involving a low-resource language: Google automl vs. from-scratch nmt systems. *Translating and the Computer*, 41:113–124.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Ha Nguyen Tien and Huyen Nguyen Thi Minh. 2019. Long sentence preprocessing in neural machine translation. In *2019 IEEE-RIVF International Conference on Computing and Communication Technologies (RIVF)*, pages 1–6. IEEE.

Mengzhou Xia, Xiang Kong, Antonios Anastasopoulos, and Graham Neubig. 2019. Generalized data augmentation for low-resource translation. *arXiv preprint arXiv:1906.03785*.

Author Index