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. ©2020 NLP Association of India (NLPAI)

(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) Valvalidation 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*.