LTRC-MT Simple & Effective Hindi-English Neural Machine Translation Systems at WAT 2019

Vikrant Goyal
IIIT Hyderabad
vikrant.goyal@research.iiit.ac.in

Dipti Misra Sharma
IIIT Hyderabad
dipti@iiit.ac.in

Abstract

This paper describes the Neural Machine Translation systems of IIIT-Hyderabad (LTRC-MT) for WAT 2019 Hindi-English shared task. We experimented with both Recurrent Neural Networks & Transformer architectures. We also show the results of our experiments of training NMT models using additional data via backtranslation.

1 Introduction

Neural Machine Translation (Luong et al., 2015; Bahdanau et al., 2014; Johnson et al., 2017; Wu et al., 2017; Vaswani et al., 2017) has been receiving considerable attention in the recent years, given its superior performance without the demand of heavily hand crafted engineering efforts. NMT often outperforms Statistical Machine Translation (SMT) techniques but it still struggles if the parallel data is insufficient like in the case of Indian languages. Hindi being one of the most common spoken Indian languages, continue to remain as a low resource language demanding further attention from the research community. The Hindi-English pair has limited availability of sentence level aligned bitext as parallel corpora.

This paper describes an overview of the submission of IIIT Hyderabad (LTRC) in WAT 2019 (Nakazawa et al., 2019) Hindi-English Machine Translation shared task. We experimented with both attention-based LSTM encoder-decoder architecture & the recently proposed Transformer architecture. We used Byte Pair Encoding (BPE) to enable open vocabulary translation. We then leveraged synthetic data generated by our own models to improve the translation performance.

2 Neural MT Architecture

In this section, we briefly discuss the attention-based LSTM encoder-decoder architecture & the Transformer model.

2.1 Attention-based encoder-decoder

In this architecture, the NMT model consists of an encoder and a decoder, each of which is a Recurrent Neural Network (RNN) as described in (Luong et al., 2015). The model directly estimates the posterior distribution $P_\theta(y|x)$ of translating a source sentence $x = (x_1, .., x_n)$ to a target sentence $y = (y_1, .., y_m)$ as:

$$P_\theta(y|x) = \prod_{t=1}^{m} P_\theta(y_t|y_1, y_2, .., y_{t-1}, x) \quad (1)$$

Each of the local posterior distribution $P(y_t|y_{1,2}, .., y_{t-1}, x)$ is modeled as a multinomial distribution over the target language vocabulary which is represented as a linear transformation followed by a softmax function on the decoder's output vector $\tilde{h}_t^{dec}$:

$$c_t = AttentionFunction(h_{1:n}^{enc}; h_t^{dec}) \quad (2)$$

$$\tilde{h}_t^{dec} = tanh(W_o[h_t^{dec}; c_t]) \quad (3)$$

$$P(y|y_1, y_2, .., y_{t-1}, x) = softmax(W_s\tilde{h}_t^{dec}; \tau) \quad (4)$$

where c_t is the context vector, h^{enc} and h^{dec} are the hidden vectors generated by the encoder and decoder respectively, AttentionFunction(. , .) is the attention mechanism as shown in (Luong et al., 2015) and [. ; .] is the concatenation of two vectors.

An RNN encoder first encodes x to a continuous vector, which serves as the initial hidden vector for the decoder and then the decoder performs recursive updates to produce a sequence of hidden vectors by applying the transition function f as:

$$h_t^{dec} = f(h_{t-1}^{dec}, [\tilde{h}_{t-1}^{dec}; e(y_t)]) \quad (5)$$

where e(.) is the word embedding operation. Popular choices for mapping f are Long-Short-Term

Proceedings of the 6th Workshop on Asian Translation, pages 137–140
Hong Kong, China, November 4, 2019.

Memory (LSTM) units and Gated Recurrent Units (GRU), the former of which we use in our models.

An NMT model is typically trained under the maximum log-likelihood objective:

$$\max_{\theta} J(\theta) = \max_{\theta} \mathrm{E}_{(x,y)\sim D}[\log P_\theta(y|x)] \quad (6)$$

where D is the training set. Our NMT model uses a bi-directional LSTM as an encoder and a uni-directional LSTM as a decoder with global attention (Luong et al., 2015) .

2.2 Transformer Model

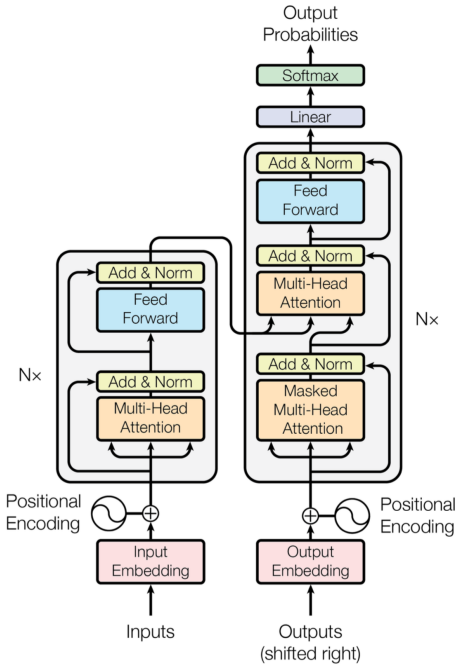

Figure 1: Transformer model architecture from Vaswani et al. (2017)

The Transformer (Vaswani et al., 2017) model is the first NMT model relying completely on self-attention mechanism to compute representations of its input and output without using recurrent neural networks (RNN) or convolutional neural networks (CNN). RNNs read one word at a time, having to perform multiple steps before generating an output that depends on words that are far away. But it has been shown that the more steps required, the harder it is for the network to learn to make these decisions (Bahdanau et al., 2014). RNNs being sequential in nature, do not effectively exploit the modern computing devices such as GPUs which rely on parallel processing. The Transformer is also an encoder-decoder model that was conceived to solve these problems. The encoder is composed of three stages. In the first stage input words are projected into an embedded vector space. In order to capture the notion of token position within the sequence, a positional encoding is added to the embedded input vectors. The second stage is a multi-headed self-attention. Instead of computing a single attention, this stage computes multiple attention blocks over the source, concatenates them and projects them linearly back onto a space with the initial dimensionality. The individual attention blocks compute the scaled dot-product attention with different linear projections. Finally a position-wise fully connected feed-forward network is used, which consists of two linear transformations with a ReLU activation (Nair and Hinton, 2010) in between.

The decoder operates similarly, but generates one word at a time, from left to right. It is composed of five stages. The first two are similar to the encoder: embedding and positional encoding and a masked multi-head self-attention, which unlike in the encoder, forces to attend only to past words. The third stage is a multi-head attention that not only attends to these past words, but also to the final representations generated by the encoder. The fourth stage is another position-wise feed-forward network. Finally, a softmax layer allows to map target word scores into target word probabilities. For more specific details about the architecture, refer to the original paper (Vaswani et al., 2017).

2.3 Subword Segmentation for NMT

Neural Machine Translation relies on first mapping each word into the vector space, and traditionally we have a word vector corresponding to each word in a fixed vocabulary. Addressing the problem of data scarcity and the hardness of the system to learn high quality representations for rare words, (Sennrich et al., 2015b) proposed to learn subword units and perform translation at a subword level. With the goal of open vocabulary NMT, we incorporate this approach in our system as a preprocessing step. In our early experiments, we note that Byte Pair Encoding (BPE) works better than UNK replacement techniques & also aids in better translation performance. For all of our systems, we learn separate vocabularies for Hindi and English each with 32k merge operations. With the help of BPE, the vocabulary size is reduced

drastically and we no longer need to prune the vocabularies. After the translation, we do an extra post processing step to convert the target language subword units back to normal words. We found this approach to be very helpful in handling rare word representations.

2.4 Synthetic Training Data

To utilize monolingual data along with IITB corpus, we employ back translation. Backtranslation (Sennrich et al., 2015a) is a widely used data augmentation technique for aiding Neural Machine Translation for languages low on parallel data. The method works by generating synthetic data on the source side from target side monolingual data using a target-to-source NMT model. The synthetic parallel data thus formed is combined with the actual parallel data to train a new NMT model. We used around 10M English sentences and backtranslated them into Hindi using a English-Hindi NMT model.

3 Experimental Setup

3.1 Dataset

In our experiments, we used IIT-Bombay (Kunchukuttan et al., 2017) Hindi-English parallel data provided by the organizers. The training corpus provided by the organizers, consists of data from mixed domains. There are roughly 1.5M samples in training data from diverse sources, while the development and test sets are from news domains. In addition to this, around 10M English monolingual data from WMT14 newscrawl articles is used in our backtranslation enabled attempts at training an NMT system.

Table 1: Statistics of our processed parallel data.

Dataset	Sentences	Tokens
IITB Train	15,28,631	21.5M / 20.3M
IITB Test	2,507	62.3k / 55.8k
IITB Dev	520	9.7k / 10.3k

3.2 Data Processing

We used Moses (Koehn et al., 2007) toolkit for tokenization and cleaning the English side of the data. Hindi side of the data is first normalized with Indic NLP library[1] followed by tokenization with the same library. As our preprocessing step, we removed all the sentences of length greater than 80 from our training corpus. We used BPE segmentation with 32k merge operations. During training, we lowercased all of our training data & used truecase[2] as a truecaser during testing.

3.3 Training Details

For all of our experiments, we used OpenNMT-py (Klein et al., 2018) toolkit. We used both attention-based LSTM models and Transformer models in our submissions.

We used an LSTM based Bi-directional encoder and a unidirectional decoder along with global attention mechanism. We kept 4 layers in both the encoder & decoder with embedding size set to 512. The batch size was set to 64 and a dropout rate of 0.3. We used Adam optimizer (Kingma and Ba, 2014) for all our experiments.

For our transformer model, we used 6 layers in both encoder and decoder with 512 hidden units in each layer. The word embedding size was set to 512 with 8 heads. The training is run in batches of maximum 4096 tokens at a time with dropout set to 0.3. The model parameters are optimized using Adam optimizer.

4 Results

In table 2, we report Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002) score, Rank-based Intuitive Bilingual Evaluation Score (RIBES) (Isozaki et al., 2010), Adequacy-fluency metrics (AM-FM) (Banchs et al., 2015) and the Human Evaluation results provided by WAT 2019 for all our attempts. The results show that our NMT system based on Transformer & backtranslation is ranked 2nd among all the constraint submissions made in WAT 2019 Hindi-English shared task & is ranked 3rd overall.

5 Conclusion Future Work

We believe that NMT is indeed a promising approach for Machine Translation of low resource languages. In this paper, we showed the effectiveness of Transformer models on a low resource languages pair Hindi-English. Additionally we show, how synthetic data can help improving the NMT systems for Hindi-English.

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

[2] https://pypi.org/project/truecase/

Table 2: This table describes the results of WAT 2019 evaluation of our submitted systems & compared with the best system submissions of WAT 2019 & the previous year. 'BT' stands for backtranslation.

Architecture	BLEU	RIBES	AM-FM	Human
Transformer	16.32	0.729072	0.563590	-
LSTM with global attention + BT	17.07	0.729059	0.587060	-
Transformer + BT	**18.64**	**0.735358**	**0.594770**	**3.43**
2018 Best	17.80	0.731727	0.611090	2.96
2019 Best (Constraint)	19.06	0.741197	0.566490	3.83
2019 Best (Unconstraint)	22.91	0.768324	0.641730	4.14

References

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

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

Hideki Isozaki, Tsutomu Hirao, Kevin Duh, Katsuhito Sudoh, and Hajime Tsukada. 2010. Automatic evaluation of translation quality for distant language pairs. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 944–952. Association for Computational Linguistics.

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

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

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

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the association for computational linguistics companion volume proceedings of the demo and poster sessions*, pages 177–180.

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

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

Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814.

Toshiaki Nakazawa, Chenchen Ding, Raj Dabre, Hideya Mino, Isao Goto, Win Pa Pa, Nobushige Doi, Yusuke Oda, Anoop Kunchukuttan, Shantipriya Parida, Ondej Bojar, and Sadao Kurohashi. 2019. Overview of the 6th workshop on Asian translation. In *Proceedings of the 6th Workshop on Asian Translation*, Hong Kong. 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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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*.

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.

Lijun Wu, Yingce Xia, Li Zhao, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2017. Adversarial neural machine translation. *arXiv preprint arXiv:1704.06933*.