Comparing Recurrent and Convolutional Architectures for English-Hindi Neural Machine Translation

Sandhya Singh, Ritesh Panjwani, Anoop Kunchukuttan, Pushpak Bhattacharyya
Center for Indian Language Technology
Department of Computer Science & Engineering
Indian Institute of Technology Bombay
{sandhya, ritesh, anoopk, pb}@cse.iitb.ac.in

Abstract

In this paper, we empirically compare the two encoder-decoder neural machine translation architectures: convolutional sequence to sequence model (ConvS2S) and recurrent sequence to sequence model (RNNS2S) for English-Hindi language pair as part of IIT Bombay's submission to WAT2017 shared task. We report the results for both English-Hindi and Hindi-English direction of language pair.

1 Introduction

Neural Machine Translation (NMT) systems are currently being widely investigated in the research community due to the benefits of distributed representation and continuous space modeling in generating more fluent outputs. In this paper, we report the results of our experiments with NMT for English-Hindi language pair for the shared task in the 4th Workshop on Asian Translation (Nakazawa et al., 2017). Hindi is the most widely spoken language in the Indian subcontinent, while English is a major link language in India as well across the world. Hence, English-Hindi is an important language pair for machine translation.

In this work, we focus on comparing two variants of the encoder decoder architectures. Section 2 describes our systems. Section 3 describes the experimental setup. Section 4 describes the results and observations of our experiments. Section 5 concludes the report.

2 System Description

We trained Neural Machine Transaltion systems using the encoder-decoder architecture with attention (Bahdanau et al., 2014) for English-Hindi as well Hindi-English translation. We compared convolutional neural network (ConvS2S) (Gehring et al., 2017) and recurrent neural network (RNNS2S) (Bahdanau et al., 2014) based sequence to sequence learning architectures. While RNN based architectures have proved to be successful and produce state-of-the-art results for machine translation, they take a long time to train. The temporal dependencies between the elements in the sequence due to the RNN state vector requires sequential processing. On the other hand, different parts of the sequence can be processed in parallel using a ConvS2S. Hence, it is appealing to explore ConvS2S as the basis of an architecture to speed up training and decoding. Recent work (Gehring et al., 2017) has shown that a purely CNN based encoder-decoder network is competitive with a RNN based network.

2.1 Recurrent sequence to sequence model (RNNS2S)

Recurrent sequence to sequence model (Bahdanau et al., 2014) is currently the most popular method for neural machine translation. It is been shown to be useful for other sequence to sequence tasks like image captioning (Vinyals et al., 2015), language modeling, question answering (Wang and Nyberg, 2015) *etc*. The typical architecture encodes the sequence of source word embeddings to generate annotations for the source words. The encoder is typically a bi-directional RNN layer of LSTM or GRU units. The final state of the encoder is used to initialize the decoder. The decoder is also an RNN which generates one output token at a time. Each output token is predicted based on the decoder state, previous output word and the context vector. The context vector encodes source information required for predicting the words, and is generated using an attention mechanism on the source word annotations. Please refer to Bahdanau et al. (2014) for an detailed description of the method.

Proceedings of the 4th Workshop on Asian Translation, pages 167–170,
Taipei, Taiwan, November 27, 2017. ©2017 AFNLP

2.2 Convolutional sequence to sequence model (ConvS2S)

In convolutional sequence to sequence model (Gehring et al., 2017), the input sequence is encoded into distributional vector space using a CNN and decoded back to output sequence again using CNN instead of RNN (Sutskever et al., 2014). Each input element embedding is combined with its positional embedding (signifies the position of the input element). Positional embeddings help the network to realize what part of input it is dealing with, currently.

Encoder-Decoder. Both the encoder and decoder are CNN blocks along with a multistep attention mechanism with multiple 'hops' (Sukhbaatar et al., 2015). Each block consists of one dimensional convolutions followed by a Gated Linear Unit (GLU) non-linearity (Dauphin et al., 2016). GLU is a gating function over the outputs of the convolutions. The multi-step attention mechanism suggests that the attention mechanism is applied to every layer in the decoder. The attention of the first layer gives contextual information which is then given as an input to the next layer that considers this information while calculating the attention weights of the current layer.

Set	# Sentences	# Tokens	
		En	Hi
Train	1,492,827	20,666,365	22,164,816
Test	2,507	49,394	57,037
Development	520	10,656	10,174

Table 1: Statistics of data sets

Method	BLEU	RIBES	AMFM	HUMAN
RNNS2S	11.55	0.6829	0.5570	21
ConvS2S	**13.76**	**0.6975**	-	-

Table 2: Hindi to English Translation

Method	BLEU	RIBES	AMFM	HUMAN
RNNS2S	**12.23**	0.6886	0.6248	28.75
ConvS2S	11.73	**0.6903**	-	-

Table 3: English to Hindi Translation

3 Experimental Setup

3.1 Data

The data for WAT2017 shared task for English-Hindi language is a mix domain data collected from different sources at CFILT[1]lab. The data provided was in tokenized format using moses tokenizer for English side and Indic NLP library[2]for Hindi side of the parallel data. The training data was further cleaned for a sentence length of 100 words. Table-1 shows data statistics used for the experiments.

Encoder	Decoder	BLEU	
		En-Hi	Hi-En
4	3	7.84	8.67
9	5	11.43	13.05
13	7	**11.73**	**13.76**

Table 4: Different number of encoder and decoder layers in ConvS2S in terms of BLEU.

3.2 Training

The RNNS2S model was trained using Nematus[3] framework. To handle rare words, subword[4] technique was used through byte pair encoding(BPE) Shibata et al. (1999) with 16000 BPE operations. Since there is no similarity between English and Hindi language vocabulary, both the languages were trained separately for BPE. The encoder and decoder hidden layer size was kept at 512 and word embedding size as 256. The model was trained with a batch size of 40 sentences and maximum sentence length of 100 using AdaDelta (Zeiler, 2012) optimizer with a learning rate of 0.0001 and no dropout setting. The output parameters were saved after every 10000 iterations. The decoding was done using a beam size of 12 and ensemble of last 3 models and the best model taken together.

The ConvS2S model was trained using Fairseq[5], an open source library developed by Facebook for neural machine translation using CNN or RNN networks. For handling the rare words, the source side and target side corpora were segmented using byte pair encoding (BPE) (Shibata et al., 1999). The baseline model with 4 encoder layers and 3 decoder layers was trained using *nag* optimizer (Gehring et al., 2017) with a learning rate of 0.25 with 0.2 as its dropout value and gradient clipping was also applied.

[1] http://www.cfilt.iitb.ac.in/
[2] http://anoopkunchukuttan.github.io/indic_nlp_library/
[3] https://github.com/EdinburghNLP/nematus
[4] https://github.com/rsennrich/subword-nmt
[5] https://github.com/facebookresearch/fairseq

Team	BLEU	RIBES	AM-FM	Pairwise	Adequacy
2016 Best	18.72	71.68	67.07	57.25	3.36
XMUNLP	21.39	74.97	68.88	64.5	3.86
IITB-MTG (RNNS2S)	12.23	68.86	62.48	28.75	2.68
IITB-MTG (ConvS2S)	11.73	69.03	-	-	-

Table 5: English to Hindi Translation Systems at WAT2017

Team	BLEU	RIBES	AM-FM	Pairwise	Adequacy
XMUNLP	22.44	75.09	62.95	68.25	3.51
IITB-MTG (RNNS2S)	11.55	68.29	55.7	21	2.29
IITB-MTG (ConvS2S)	13.76	69.75	-	-	-

Table 6: Hindi to English Translation Systems at WAT2017

The inferencing was done using beam search with a beam size of 10 for both Hindi-English and English-Hindi translation task. The model was also trained with more number of layers in the encoder and the decoder. The resulting BLEU scores for different number of encoder and decoder layers are shown in Table 4. The best results were obtained when the number of encoder layers were set to 13 and decoder layers to 7, with learning rate of 0.1 and no dropout regularization. The resulting BLEU scores with this setting for Hindi-English and English-Hindi are shown in Table 2 and Table 3 respectively.

4 Results and Observation

The Table 2 and the Table 3 shows the different evaluation metrics such as Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002), Rank-based Intuitive Bilingual Evaluation Score (RIBES) (Group et al., 2013), Adequacy-Fluency Metrics (AMFM) (Banchs et al., 2015) (N/A for ConvS2S model) and human evaluation score (HUMAN) (N/A for ConvS2S model) for Hindi-English and English-Hindi translation pairs.

In Hindi to English translation, the ConvS2S model outperforms the RNNS2S model in terms of BLEU score and the RIBES score. On the other hand, in English to Hindi translation, the RNNS2S model performs better than the ConvS2S model in terms of BLEU score and the RIBES score is at par with the ConvS2S model.

The JPO Adequacy and pairwise evaluation of our RNNS2S output was compared against WAT2016 best system. Table 5 and table 6 show the evaluation results of all other systems in comparison to our submission. The results clearly indicate the scope of fine tuning our system parameters. Due to time constraint, the ConvS2S output could not be submitted for manual evaluation. But the increasing trend of BLEU Scores have motivated us to continue our experimentation for a deeper analysis.

Further experimentation is required to see if the ConvS2S can perform better on English-Hindi as well. One way to test this is by increasing the number of encoder and/or decoder layers even further. This is because, in the Table 4 we can clearly observe that the BLEU scores increases when number of encoder and decoder layers are increased. More experiments are required with RNNS2S architecture as well.

5 Conclusion

In our system submission, we compared two sequence to sequence architectures: RNN based and CNN based for the English-Hindi language pairs. The BLEU scores of CNN architecture improves by further tunning the parameters.

In future, we would like to investigate the threshold of hyperparameters for RNNS2S and ConvS2S architectures for this language pair keeping processing time in consideration.

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 D'Haro, and Haizhou Li. 2015. Adequacy-fluency metrics: Evaluating mt in the continuous space model framework. *IEEE*

Transactions on Audio, Speech, and Language Processing, 23(3):472–482.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2016. Language modeling with gated convolutional networks. *arXiv preprint arXiv:1612.08083*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

Linguistic Intelligence Research Group et al. 2013. Ntt communication science laboratories. ribes: Rank-based intuitive bilingual evaluation score.

Toshiaki Nakazawa, Shohei Higashiyama, Chenchen Ding, Hideya Mino, Isao Goto, Graham Neubig, Hideto Kazawa, Yusuke Oda, Jun Harashima, and Sadao Kurohashi. 2017. Overview of the 4th workshop on asian translation. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*, Taipei, Taiwan.

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.

Yusuxke Shibata, Takuya Kida, Shuichi Fukamachi, Masayuki Takeda, Ayumi Shinohara, Takeshi Shinohara, and Setsuo Arikawa. 1999. Byte pair encoding: A text compression scheme that accelerates pattern matching. Technical report, Technical Report DOI-TR-161, Department of Informatics, Kyushu University.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

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.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, volume 2, pages 707–712.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.