A Preordered RNN Layer Boosts Neural Machine Translation in Low Resource Settings

Mohaddeseh Bastan
Stony Brook University
mbastan@cs.stonybrook.edu

Shahram Khadivi*
Amirkabir University of Technology
khadivi@aut.ac.ir

Abstract

Neural Machine Translation (NMT) models are strong enough to convey semantic and syntactic information from the source language to the target language. However, these models are suffering from the need for a large amount of data to learn the parameters. As a result, for languages with scarce data, these models are at risk of underperforming. We propose to augment attention based neural network with reordering information to alleviate the lack of data. This augmentation improves the translation quality for both English to Persian and Persian to English by up to 6% BLEU absolute over the baseline models.

1 Introduction

NMT has recently shown promising results in machine translation (Wu et al., 2016; Luong et al., 2015; Bastan et al., 2017). In statistical machine translation (SMT), the problem is decomposed into sub-models and each individual model is trained separately, while NMT is capable of training an end-to-end model. For instance, in SMT the reordering model is a feature that is trained separately and is used jointly with other features to improve the translation, while in NMT it is assumed that the model will learn the order of the words and phrases itself.

Sequence-to-sequence NMT models consist of two parts, an encoder to encode the input sequence to the hidden state and a decoder that decodes the hidden state to get the output sequence (Cho et al., 2014; Bahdanau et al., 2014). The encoder model is a bidirectional Recurrent Neural Network (RNN), the source sentence is processed once from the beginning to the end and once in parallel from the end to the beginning. One of the ideas that have not been well-explored in NMT so far is the use of existing reordering models in SMT. We propose to add another layer to the encoder that includes reordering information. The intuition behind our proposal comes from the improvement achieved by bidirectional encoder model. If processing the source sentence in both directions help sequence-to-sequence model to learn better representation of the context in hidden states, adding the order of the input words as they are appearing in the output sequence as another layer may also help the model to learn a better representation in both context vectors and hidden states. In this paper we investigate this hypothesis that another layer in the encoder to process a preordered sentence can outperform both encoder architecture with two or three RNN layers. We empirically show in the experiments that adding the reordering information to NMT can improve the translation quality when we are in shortage of data.

There are a few attempts to improve the SMT using neural reordering models (Cui et al., 2015; Li et al., 2014, 2013; Aghasadeghi and Bastan, 2015). In Zhang et al. (2017), three distortion models have been studied to incorporate the word reordering knowledge into NMT. They used reordering information to mainly improve the attention mechanism.

In this paper, we are using a soft reordering model to improve the bidirectional attention based NMT. This model consists of two different parts. The first part is creating the soft reordering information using the input and output sequence, the second part is using this information in the attention based NMT.

The rest of the paper is as follow, in section 2 a review of sequence-to-sequence NMT is provided, in section 3 the preordered model is proposed, section 4 explains the experiments and results, and section 5 concludes the paper.

* This work is done in 2017 when Shahram Khadivi was with Amirkabir University of Technology.

Proceedings of the 29th International Conference on Computational Linguistic, pages 93–98
October 12–17, 2022.

2 Sequence-to-Sequence NMT

Bahdanau et al. (2014) proposed a joint translation and alignment model which can both learn the translation and the alignment between the source and the target sequence. In this model the decoder at each time step, finds the maximum probability of the output word y_i given the previous output words $y_1, ..., y_{i-1}$ and the input sequence X as follow:

$$p(y_i|y_1, ..., y_{i-1}, X) = softmax(g(y_{i-1}, s_i, c_i)) \quad (1)$$

Where X is the input sequence, g is a nonlinear function, s_i is the hidden state, and c_i the context vector using to predict output i. s_i is the hidden state at the current step which is defined as follow:

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad (2)$$

The notation c_i is the context vector for output word y_i. The context vector is the weighted sum of the hidden states as follow:

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j \quad (3)$$

The weights in this model are normalized outputs of the alignment model which is a feed-forward neural network. It uses s_{i-1} and h_j as input and outputs a score e_{ij}. This score is then normalized and used as the weight for computing the context vector as follow:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T} (\exp e_{ik})} \quad (4)$$

In the encoder, a bidirectional neural network is used to produce the hidden state h. For each input word x_i there is a forward and a backward hidden state computed as follow respectively:

$$\overrightarrow{h_i} = \overrightarrow{f}(\overrightarrow{h_{i-1}}, x_i) \quad (5)$$

$$\overleftarrow{h_i} = \overleftarrow{f}(\overleftarrow{h_{i-1}}, x_i) \quad (6)$$

Forward and backward hidden states are then concatenated to produce the hidden state h_i as follow:

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}] \quad (7)$$

3 Preordered RNN

The attention-based model is able to address some of the shortcomings of the simple encoder-decoder model for machine translation. It works fine when we have plenty of data. But if we are in lack of data the attention-based model suffers from lack of information for tuning all the parameters. We can use some other information of the input data to inject into the model and get even better results. In this paper, a model is proposed using reordering information of the data set to address the issue of shortage of data. Adding this information to the model, it can improve the attention-based NMT significantly.

3.1 Building Soft Reordered Data

Adding a preordered layer to the encoder of the sequence model boosts the translation quality. This layer add some information to the model which previously hasn't been seen. The preordered data is the source sentence which is reordered using the information in target sentence. The reordered models have been used in statistical machine translation and they could improve the translation quality (Visweswariah et al., 2011; Tromble and Eisner, 2009; Khalilov et al., 2010; Collins et al., 2005; Xia and McCord, 2004).
To obtain the soft reordering model, we first need to have the word alignment between the source and the target sentences, then by using heuristic rules we change the alignment to reordering. The reordered sequence model is built upon the alignment model. First by using GIZA++ (Och and Ney, 2003) the alignment model between the input sequence and output sequence is derived. The main difference between reordering and alignment is that alignment is a many-to-many relation, while the reordering is a one-to-one relation. It means one word in the input sequence can be aligned to many words in the output sequence while it can be reordered to just one position. The other difference is that the alignment is a relation from input sequence space to output sequence space while the reordering is a relation from input sequence space to itself. So we propose some heuristic rules to convert the alignment relation to the reordering relation as follow:

- If a word x in the input sequence is aligned to one and only one word y in the output sequence, the position of x in the reordering model will be the position of y.
- If a word x in the input sequence is aligned to a series of words in the output sequence, the position of x in the reordering model will be

the position of the middle word in the series[1].

- If a word in the input sequence is not aligned to any word in the output sequence, the position for that word is the average positions of the previous and the next word.

These heuristic rules are inspired by the rules which have been proposed in Devlin et al. (2014). The difference is that they are trying to align one and only one input word to all output words, but we are trying to align each word in the input sequence to one and only one position in the same space.

The order of applying these rules is important. We should apply the first rule, then the second rule and finally the third rule to all possible words. If a word is aligned to a position but that position is full, we align it to the nearest empty position. We arbitrarily prioritize the left position to the right position whenever they have the same priority. At the end, each word is aligned with only one position, but there may be some positions which are empty. We just remove the empty positions between words to map the sparse output space to the dense input space. We can build the reordered training data using these rules and use them for training the model. In the next section, we see how the reordered data is used in the bidirectional attention based NMT.

3.2 Three-layer Encoder

The bidirectional encoder has two different layers. The first layer consists of the forward hidden states built by reading the input sequence from left to right and the second layer consists of the backward hidden states, built by reading the input sequence from right to left. We add another hidden layer to the encoder which is built by reading the input sequence in the reordered order. We build the hidden layer of the reordered input as follow:

$$hr_i = f(hr_{i-1}, xr_i) \tag{8}$$

Here xr_i is the word in position i of the reordered data and hr_i is the hidden representation of x_i in reordered set. The function for computing hr is the same as in equation 5 and 6. Then the hidden representation h is computed by concatenating the forward hidden layer, backward hidden layer and reordering hidden layer as follow:

$$h_i = [\overrightarrow{h_i}, \overleftarrow{h_i}, hr_i] \tag{9}$$

[1]We arbitrary round down the even number. For example, the middle position between 1,3,5,7 is the 3rd position.

Corpus	#sents	#words	
		English	Persian
Training	26142	264235	242154
Development	276	3442	3339
Test	250	2856	2668

Table 1: The statistics of data set

4 Experiments

The proposed model has been evaluated on English-Persian translation data set. We believe that adding the reordering information results in a better model in case of low resource data. We evaluate the translation quality based on BLEU (Papineni et al., 2002) and TER (Snover et al., 2006). For implementation we use the Theano (Bergstra et al., 2011) framework.

4.1 Dataset

We use Verbmobil (Bakhshaei et al., 2010), an English-Persian data set, this data set can show the effectiveness of the model on scarce data resources. The detailed information of the data set is provided in 1. In this table, the number of words, shown with #words, number of sentences in each corpus is shown in column #sents.

4.2 Baseline

The baseline model for our experiments is the bidirectional attention based neural network (Bahdanau et al., 2014) as explained in section 1. There are various papers to improve the basic attention based decoder of the baseline, among all we used guided alignment (Chen et al., 2016).

4.3 Reordering Development and Test Set

For building the reordered training set, we use alignment model and heuristic rules. For development and test set, as we don't have access to the target language, we use a preordering algorithm proposed in Nakagawa (2015). This algorithm is the improved version of preordering algorithm based on Bracketing Transduction Grammar (BTG). Briefly, this algorithm builds a tree based on the words, so that each node has a feature vector and a weight vector. Among all possible trees on the data set, the tree with maximum value for the weighted sum of the feature vectors is chosen

Reordering Method			
Training Set	Dev/Test Set	BLEU	TER
HG	BI	30.53	53.25
BI	BI	27.91	56.68
BG	BG	25.93	58.1

Table 2: The comparison between different reordering methods on Verbmobil data. HG means the data reordered using alignment model with GDFA and heuristic rules, BI and BG means the data is reordered on intersection alignment and GDFA alignment, respectively, both using (Nakagawa, 2015) algorithm.

as reordering tree. Using a projection function, the tree is converted into the reordered output.
This algorithm also needs part of speech (POS) tagger and word class. For Persian POS tagging we use CMU NLP Farsi tool (Feely et al., 2014) and for the English POS tagging, we use Stanford POS tagger (Toutanova et al., 2003). For word class we use the GIZA ++ word class which is an output of creating alignment.

4.4 Results

We analyzed our model with different configurations. First we use different methods to reorder training, development and test set. The results are shown in 2. In this table, the best results of different combinations for building reordered data is shown. HG means for building the reordered data, heuristic rules and alignment with GDFA (Koehn, 2005) is used. BI means the algorithm in (Nakagawa, 2015) and alignment with intersection method is used to build the reordered data, BG means alignment with GDFA and reordering algorithm in (Nakagawa, 2015) is used. The best possible combinations are shown in Table 2.

In Table 3 we can compare the best 3-layer network with two different 2-layer networks. The 3-layer network has apparently three layers in the encoder, the first two layers are the forward and the backward RNNs, the third layer is again an RNN trained either on the reordered source sentence or the original sentence. The 2-layer network refers to the bidirectional attention based NMT as described in Section 2. This model id trained once with the original sentence, and once with the reordered sentence. As we see, reordering the input can improve the model. It shows that the information we are adding to our model is useful. So using the best 3layer model can use both information of reordering and information of the ordered

Reordering Method			
Data set	Model	BLEU	TER
En → Pr	Baseline SMT	30.47	–
	Baseline NMT	27.42	50.78
	3-layer RpL	27.58	50.04
	2-layer RI	29.6	50.96
	3-layer RL	31.03	47.5
	Ensemble	**32.74**	**46.4**
Pr → En	Baseline SMT	26.91	–
	Baseline NMT	26.12	55.87
	3-layer RpL	26.38	57.42
	2-layer RI	27.52	54.12
	3-layer RL	30.53	53.25
	Ensemble	**32.17**	**52.12**

Table 3: The comparison between different models. base line in SMT is the result of translation in statistical machine translation. The base line NMT is the bidirectional attention based neural network using guided alignment (Bahdanau et al., 2014; Chen et al., 2016). The 2layer RI is the basic model with reordered input. The 3layer RL is the model proposed in this paper. The 3layer RpL is a 3layer model with two forward and one backward layers (No reordering layer). The ensemble model is the combination of different models.

data, so it can improve the translation model significantly. Also we see that adding just a simple repeated layer to bidirectional encoder, can improve the model. But not as much as the reordered layer. Finally, the ensemble of different models has the best results.
There are different interpretations behind this results. Because NMT has too many parameters, it is difficult for scarce data to learn all of the parameters correctly. So adding explicit information using the same data can help the model to learn the parameters better. In addition, although we expect that all the statistical features we use in SMT automatically be trained in NMT, but it can not learn them as well as SMT.

5 Conclusion

In this paper we analyzed adding reordering information to NMTs. NMTs are strong because they can translate the source language into target without breaking the problem into sub problems. In this paper we proposed a model using explicit information which covers the hidden feature like reordering. The improvements is the result of adding extra information to the model, and helping the neural network learn the parameters in case of scarce data better.

References

Amir Pouya Aghasadeghi and Mohadeseh Bastan. 2015. Monolingually derived phrase scores for phrase based smt using neural networks vector representations. *arXiv preprint arXiv:1506.00406*.

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

Somayeh Bakhshaei, Shahram Khadivi, and Noushin Riahi. 2010. Farsi-german statistical machine translation through bridge language. In *Telecommunications (IST), 2010 5th International Symposium on*, pages 557–561. IEEE.

Mohaddeseh Bastan, Shahram Khadivi, and Mohammad Mehdi Homayounpour. 2017. Neural machine translation on scarce-resource condition: a case-study on persian-english. In *2017 Iranian Conference on Electrical Engineering (ICEE)*, pages 1485–1490. IEEE.

James Bergstra, Frédéric Bastien, Olivier Breuleux, Pascal Lamblin, Razvan Pascanu, Olivier Delalleau, Guillaume Desjardins, David Warde-Farley, Ian Goodfellow, Arnaud Bergeron, et al. 2011. Theano: Deep learning on gpus with python. In *NIPS 2011, BigLearning Workshop, Granada, Spain*, volume 3. Citeseer.

Wenhu Chen, Evgeny Matusov, Shahram Khadivi, and Jan-Thorsten Peter. 2016. Guided alignment training for topic-aware neural machine translation. *arXiv preprint arXiv:1607.01628*.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics.

Yiming Cui, Shijin Wang, and Jianfeng Li. 2015. Lstm neural reordering feature for statistical machine translation. *arXiv preprint arXiv:1512.00177*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1370–1380.

Weston Feely, Mehdi Manshadi, Robert E Frederking, and Lori S Levin. 2014. The cmu metal farsi nlp approach. In *LREC*, pages 4052–4055.

Maxim Khalilov, Khalil Sima'an, et al. 2010. Source reordering using maxent classifiers and supertags. In *Proc. of EAMT*, volume 10, pages 292–299.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for itg-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 567–577.

Peng Li, Yang Liu, Maosong Sun, Tatsuya Izuha, and Dakun Zhang. 2014. A neural reordering model for phrase-based translation. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1897–1907.

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

Tetsuji Nakagawa. 2015. Efficient top-down btg parsing for machine translation preordering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 208–218.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

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.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, volume 200.

Kristina Toutanova, Dan Klein, Christopher D Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics.

Roy Tromble and Jason Eisner. 2009. Learning linear ordering problems for better translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2*, pages 1007–1016. Association for Computational Linguistics.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil. 2011. A word reordering model for improved machine translation. In *proceedings of the conference on empirical methods in natural language processing*, pages 486–496. Association for Computational Linguistics.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th international conference on Computational Linguistics*, page 508. Association for Computational Linguistics.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1524–1534.