LSTM Neural Reordering Feature for Statistical Machine Translation

Yiming Cui, Shijin Wang and **Jianfeng Li**
iFLYTEK Research, Beijing, China
{ymcui,sjwang3,jfli3}@iflytek.com

Abstract

Artificial neural networks are powerful models, which have been widely applied into many aspects of machine translation, such as language modeling and translation modeling. Though notable improvements have been made in these areas, the reordering problem still remains a challenge in statistical machine translations. In this paper, we present a novel neural reordering model that directly models word pairs and their alignment. Further by utilizing LSTM recurrent neural networks, much longer context could be learned for reordering prediction. Experimental results on NIST OpenMT12 Arabic-English and Chinese-English 1000-best rescoring task show that our LSTM neural reordering feature is robust, and achieves significant improvements over various baseline systems.

1 Introduction

In statistical machine translation, the language model, translation model, and reordering model are the three most important components. Among these models, the reordering model plays an important role in phrase-based machine translation (Koehn et al., 2004), and it still remains a major challenge in current study.

In recent years, various phrase reordering methods have been proposed for phrase-based SMT systems, which can be classified into two broad categories:

(1) *Distance-based RM*: Penalize phrase displacements with respect to the degree of non-monotonicity (Koehn et al., 2004).

(2) *Lexicalized RM*: Conditions reordering probabilities on current phrase pairs. According to the orientation determinants, lexicalized reordering model can further be classified into word-based RM (Tillman, 2004), phrase-based RM (Koehn et al., 2007), and hierarchical phrase-based RM (Galley and Manning, 2008).

Furthermore, some researchers proposed a reordering model that conditions both current and previous phrase pairs by utilizing recursive autoencoders (Li et al., 2014).

In this paper, we propose a novel neural reordering feature by including longer context for predicting orientations. We utilize a long short-term memory recurrent neural network (LSTM-RNN) (Graves, 1997), and directly models word pairs to predict its most probable orientation. Experimental results on NIST OpenMT12 Arabic-English and Chinese-English translation show that our neural reordering model achieves significant improvements over various baselines in 1000-best rescoring task.

2 Related Work

Recently, various neural network models have been applied into machine translation.

Feed-forward neural language model was first proposed by Bengio et al. (2003), which was a breakthrough in language modeling. Mikolov et al. (2011) proposed to use recurrent neural network in language modeling, which can include much longer context history for predicting next word. Experimental results show that RNN-based language model significantly outperform standard feed-forward language model.

Proceedings of NAACL-HLT 2016, pages 977–982,
San Diego, California, June 12-17, 2016.

Devlin et al. (2014) proposed a neural network joint model (NNJM) by conditioning both source and target language context for target word predicting. Though the network architecture is a simple feed-forward neural network, the results have shown significant improvements over state-of-the-art baselines.

Sundermeyer et al. (2014) also put forward a neural translation model, by utilizing LSTM-based RNN and Bidirectional RNN. In bidirectional RNNs, the target word is conditioned on not only the history but also future source context, which forms a full source sentence for predicting target words.

Li et al. (2013) proposed to use a recursive auto-encoder (RAE) to map each phrase pairs into continuous vectors, and handle reordering problems with a classifier. Also, they suggested that by both including current and previous phrase pairs to determine the phrase orientations could achieve further improvements in accuracy (Li et al., 2014).

By far, we have noticed that this is the first time to use LSTM-RNN in reordering model. We could include much longer context information to determine phrase orientations using RNN architecture. Furthermore, by utilizing the LSTM layer, the network is able to capture much longer range dependencies than standard RNNs.

Because we need to record fixed length of history information in SMT decoding step, we only utilize our LSTM-RNN reordering model as a feature in 1000-best rescoring step. As word alignments are known after generating n-best list, it is possible to use LSTM-RNN reordering model to score each hypothesis.

3 Lexicalized Reordering Model

In traditional statistical machine translation, lexicalized reordering models (Koehn et al., 2007) have been widely used. It considers alignments of current and previous phrase pairs to determine the orientation.

Formally, when given source language sentence $f = \{f_1, ..., f_n\}$, target language sentence $e = \{e_1, ..., e_n\}$, and phrase alignment $a = \{a_1, ..., a_n\}$, the lexicalized reordering model can be illustrated in Equation 1, which only conditions on a_{i-1} and a_i, i.e. previous and current alignment.

$$p(\mathbf{o}|\mathbf{e}, \mathbf{f}) = \prod_{i=1}^{n} p(o_i|e_i, f_{a_i}, a_{i-1}, a_i) \quad (1)$$

In Equation 1, the o_i represents the set of phrase orientations. For example, in the most commonly used MSD-based orientation type, o_i takes three values: M stands for *monotone*, S for *swap*, and D for *discontinuous*. The definition of MSD-based orientation is shown in Equation 2.

$$o_i = \begin{cases} M, & a_i - a_{i-1} = 1 \\ S, & a_i - a_{i-1} = -1 \\ D, & |a_i - a_{i-1}| \neq 1 \end{cases} \quad (2)$$

For other orientation types, such as LR and MSLR are also widely used, whose definition can be found on Moses official website [1].

Recent studies on reordering model suggest that by also conditioning previous phrase pairs can improve context sensitivity and reduce reordering ambiguity.

4 LSTM Neural Reordering Model

In order to include more context information for determining reordering, we propose to use a recurrent neural network, which has been shown to perform considerably better than standard feed-forward architectures in sequence prediction (Mikolov et al., 2011). However, RNN with conventional back-propagation training suffers from gradient vanishing issues (Bengio et al., 1994) .

Later, the long short-term memory was proposed for solving gradient vanishing problem, and it could catch longer context than standard RNNs with sigmoid activation functions. In this paper, we adopt LSTM architecture for training neural reordering model.

4.1 Training Data Processing

For reducing model complexity and easy implementation, our neural reordering model is purely lexicalized and trained on word-level.

We will take LR orientation for explanations, while other orientation types (MSD, MSLR) can be induced similarly. Given a sentence pair and

[1] http://www.statmt.org/moses/

its alignment information, we can induce the word-based reordering information by following steps. Note that, we always evaluate the model in the order of target sentence.

(1) If current target word is one-to-one alignment, then we can directly induce its orientations, i.e. $\langle left \rangle$ or $\langle right \rangle$.

(2) If current source/target word is one-to-many alignment, then we judge its orientation by considering its first aligned target/source word, and the other aligned target/source words are annotated as $\langle follow \rangle$ reordering type, which means these word pairs inherent the orientation of previous word pair.

(3) If current source/target word is not aligned to any target/source words, we introduce a $\langle null \rangle$ token in its opposite side, and annotate this word pair as $\langle follow \rangle$ reordering type.

Figure 1 shows an example of data processing.

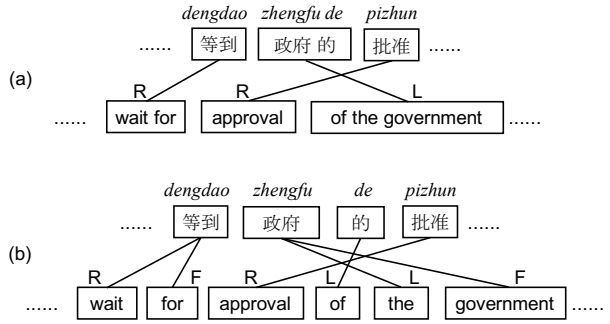

Figure 1: Illustration of data processing. (a) Original reordering (omit alignment inside each phrase); (b) processed reordering, all alignments are regularized to word level, R-right, L-left, F-follow.

4.2 LSTM Network Architecture

After processing the training data, we can directly utilize the word pairs and its orientation to train a neural reordering model.

Given a word pair and its orientation, a neural reordering model can be illustrated by Equation 3.

$$p(\mathbf{o}|\mathbf{e},\mathbf{f}) = \prod_{i=1}^{n} p(o_i|e_1^i, f_1^{a_i}, a_{i-1}, a_i) \quad (3)$$

Where $e_1^i = \{e_1, ..., e_i\}$, $f_1^{a_i} = \{f_1, ..., f_{a_i}\}$. Inclusion of history word pairs is done with recurrent neural network, which is known for its capability of learning history information.

The architecture of LSTM-RNN reordering model is depicted in Figure 2, and corresponding equations are shown in Equation 4 to 6.

$$y_i = W_1 * f_{a_i} + W_2 * e_i \quad (4)$$

$$z_i = LSTM(y_i, W_3, y_1^{i-1}) \quad (5)$$

$$p(o_i|e_1^i, f_1^{a_i}, a_{i-1}, a_i) = softmax(W_4 * z_i) \quad (6)$$

The input layer consists both source and target language word, which is in one-hot representation. Then we perform a linear transformation of input layer to a projection layer, which is also called embedding layer. We adopt extended-LSTM as our hidden layer implementation, which consists of three gating units, i.e. input, forget and output gates. We omit rather extensive LSTM equations here, which can be found in (Graves and Schmidhuber, 2005). The output layer is composed by orientation types. For example, in LR condition, the output layer contains two units: $\langle left \rangle$ and $\langle right \rangle$ orientation. Finally, we apply softmax function to obtain normalized probabilities of each orientation.

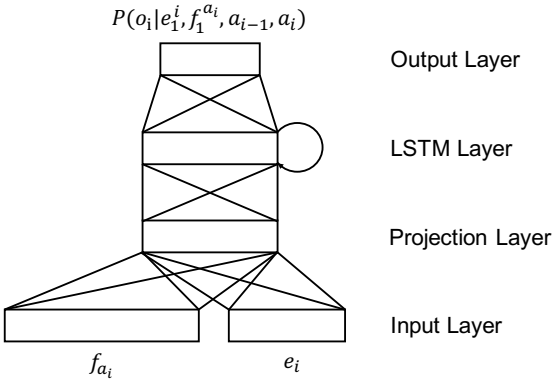

Figure 2: Architecture of LSTM neural reordering model.

5 Experiments

5.1 Setups

We mainly tested our approach on Arabic-English and Chinese-English translation. The training corpus contains 7M words for Arabic, and 4M words for Chinese, which is selected from NIST

System	Dev	Test1	Test2
Ar-En	MT04-05-06 (3795)	MT08 (1360)	MT09 (1313)
Zh-En	MT05-08 (2439)	MT08.prog (1370)	MT12.rd (820)

Table 1: Statistics of development and test set. The number of segments are indicated in brackets.

OpenMT12 parallel dataset. We use the SAMA tokenizer[2] for Arabic word tokenization, and in-house segmenter for Chinese words. The English part of parallel data is tokenized and lowercased. All development and test sets have 4 references for each segment. The statistics of development and test sets are shown in Table 1.

The baseline systems are built with the open-source phrase-based SMT toolkit Moses (Koehn et al., 2007). Word alignment and phrase extraction are done by GIZA++ (Och and Ney, 2000) with L0-normalization (Vaswani et al., 2012), and grow-diag-final refinement rule (Koehn et al., 2004). Monolingual part of training data is used to train a 5-gram language model using SRILM (Stolcke, 2002). Parameter tuning is done by K-best MIRA (Cherry and Foster, 2012). For guarantee of result stability, we tune every system 5 times independently, and take the average BLEU score (Clark et al., 2011). The translation quality is evaluated by case-insensitive BLEU-4 metric (Papineni et al., 2002). The statistical significance test is also carried out with paired bootstrap resampling method with $p < 0.001$ intervals (Koehn, 2004). Our models are evaluated in a 1000-best rescoring step, and all features in 1000-best list as well as LSTM-RNN reordering feature are retuned via K-best MIRA algorithm.

For neural network training, we use all parallel text in the baseline training. As a trade-off between computational cost and performance, the projection layer and hidden layer are set to 100, which is enough for our task (We have not seen significant gains when increasing dimensions greater than 100). We use an initial learning rate of 0.01 with standard SGD optimization without momentum. We trained model for a total of 10 epochs with cross-entropy criterion. Input and output vocabulary are set to 100K and 50K respectively, and all out-of-vocabulary words are mapped to a $\langle unk \rangle$ token.

[2] https://catalog.ldc.upenn.edu/LDC2010L01

5.2 Results on Different Orientation Types

At first, we test our neural reordering model (NRM) on the baseline that contains word-based reordering model with LR orientation. The results are shown in Table 2 and 3.

As we can see that, among various orientation types (LR, MSD, MSLR), our model could give consistent improvements over baseline system. The overall BLEU improvements range from 0.42 to 0.79 for Arabic-English, and 0.31 to 0.72 for Chinese-English systems. All neural results are significantly better than baselines ($p < 0.001$ level).

In the meantime, we also find that "Left-Right" based orientation methods, such as LR and MSLR, consistently outperform MSD-based orientations. The may caused by non-separability problem, which means that MSD-based methods are vulnerable to the change of context, and weak in resolving reordering ambiguities. Similar conclusion can be found in Li et al. (2014) .

Ar-En System	Dev	Test1	Test2
Baseline	43.87	39.84	42.05
+NRM_LR	44.43	40.53	**42.84**
+NRM_MSD	44.29	40.41	42.62
+NRM_MSLR	**44.52**	**40.59**	42.78

Table 2: LSTM reordering model with different orientation types for Arabic-English system.

Zh-En System	Dev	Test1	Test2
Baseline	27.18	26.17	24.04
+NRM_LR	**27.90**	26.58	**24.70**
+NRM_MSD	27.49	26.51	24.39
+NRM_MSLR	27.82	**26.78**	24.53

Table 3: LSTM reordering model with different orientation types for Chinese-English system.

5.3 Results on Different Reordering Baselines

We also test our approach on various baselines, which either contains word-based, phrase-based, or hierarchical phrase-based reordering model. We only show the results of MSLR orientation, which is relatively superior than others according to the results in Section 5.2.

Ar-En System	Dev	Test1	Test2
Baseline_wbe	43.87	39.84	42.05
+NRM_MSLR	44.52	40.59	42.78
Baseline_phr	44.11	40.09	42.21
+NRM_MSLR	44.52	40.73	42.89
Baseline_hier	44.30	40.23	42.38
+NRM_MSLR	44.61	40.82	42.86
Zh-En System	**Dev**	**Test1**	**Test2**
Baseline_wbe	27.18	26.17	24.04
+NRM_MSLR	27.90	26.58	24.70
Baseline_phr	27.33	26.05	24.13
+NRM_MSLR	27.86	26.46	24.73
Baseline_hier	27.56	26.29	24.38
+NRM_MSLR	28.02	26.49	24.67

Table 4: Results on various baselines for Arabic-English and Chinese-English system. "wbe": word-based; "phr": phrase-based; "hier": hierarchical phrase-based reordering model. All NRM results are significantly better than baselines ($p < 0.001$ level).

In Table 4 and 5, we can see that though we add a strong hierarchical phrase-based reordering model in the baseline, our model can still bring a maximum gain of 0.59 BLEU score, which suggest that our model is applicable and robust in various circumstances. However, we have noticed that the gains in Arabic-English system is relatively greater than that in Chinese-English system. This is probably because hierarchical reordering features tend to work better for Chinese words, and thus our model will bring little remedy to its baseline.

6 Conclusions

We present a novel work that build a reordering model using LSTM-RNN, which is much sensitive to the change of context and introduce rich context information for reordering prediction. Furthermore, the proposed model is purely lexicalized and straightforward, which is easy to realize. Experimental results on 1000-best rescoring show that our neural reordering feature is robust, and could give consistent improvements over various baseline systems.

In future, we are planning to extend our word-based LSTM reordering model to phrase-based reordering model, in order to dissolve much more ambiguities and improve reordering accuracy. Furthermore, we are also going to integrate our neural reordering model into neural machine translation systems.

Acknowledgments

We sincerely thank the anonymous reviewers for their thoughtful comments on our work.

References

Y. Bengio, P. Simard, and P. Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166.

Yoshua Bengio, Holger Schwenk, Jean Sbastien Sencal, Frderic Morin, and Jean Luc Gauvain. 2003. A neural probabilistic language model. *Journal of Machine Learning Research*, 3(6):1137–1155.

Colin Cherry and George Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June. Association for Computational Linguistics.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 176–181, Portland, Oregon, USA, June. Association for Computational Linguistics.

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)*, pages 1370–1380, Baltimore, Maryland, June. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October. Association for Computational Linguistics.

A. Graves and J. Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm networks. In *Proceedings in 2005 IEEE International Joint Conference on Neural Networks*, pages 2047–2052 vol. 4.

Alex Graves. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2004. Statistical phrase-based translation. In *Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-volume*, pages 127–133.

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain, July. Association for Computational Linguistics.

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, Seattle, Washington, USA, October. Association for Computational Linguistics.

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, Dublin, Ireland, August. Dublin City University and Association for Computational Linguistics.

T. Mikolov, S. Kombrink, L. Burget, and J. H. Cernocky. 2011. Extensions of recurrent neural network language model. In *IEEE International Conference on Acoustics, Speech Signal Processing*, pages 5528–5531.

Franz Josef Och and Hermann Ney. 2000. A comparison of alignment models for statistical machine translation. In *Proceedings of the 18th conference on Computational linguistics - Volume 2*, pages 1086–1090.

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

Andreas Stolcke. 2002. Srilm — an extensible language modeling toolkit. In *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, pages 901–904.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 14–25, Doha, Qatar, October. Association for Computational Linguistics.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA, May 2 - May 7. Association for Computational Linguistics.

Ashish Vaswani, Liang Huang, and David Chiang. 2012. Smaller alignment models for better translations: Unsupervised word alignment with the l0-norm. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–319, Jeju Island, Korea, July. Association for Computational Linguistics.