Recurrent Neural Network based Rule Sequence Model for Statistical Machine Translation

Heng Yu, Xuan Zhu

Samsung R&D Institute of China, Beijing, China
{h0517.yu, xuan.zhu}@samsung.com

Abstract

The inability to model long-distance dependency has been handicapping SMT for years. Specifically, the context independence assumption makes it hard to capture the dependency between translation rules. In this paper, we introduce a novel recurrent neural network based rule sequence model to incorporate arbitrary long contextual information during estimating probabilities of rule sequences. Moreover, our model frees the translation model from keeping huge and redundant grammars, resulting in more efficient training and decoding. Experimental results show that our method achieves a 0.9 point BLEU gain over the baseline, and a significant reduction in rule table size for both phrase-based and hierarchical phrase-based systems.

1 Introduction

Modeling long-distance dependency has always been a bottleneck for statistical machine translation (SMT). While lots of efforts have been made in solving long-distance reordering (Xiong et al., 2006; Zens and Ney, 2006; Kumar and Byrne, 2005), long-span n-gram matching (Charniak et al., 2003; Shen et al., 2008; Yu et al., 2014), much less attention has been concentrated on capturing translation rule dependency, which is not explicitly modeled in most translation systems (Wu et al., 2014).

SMT systems typically model the translation process as a sequence of translation steps, each of which uses a translation rule. These rules are usually applied independently of each other, which violates the conventional wisdom that translation should be done in context (Giménez and Màrquez, 2007). However, it is not an easy task to capture the rule dependency, which entails much longer context and more severe data sparsity. There are two major solutions: the first one is breaking the rules into bilingual word-pairs and use a n-gram translation model to incorporate lexical dependencies that span rule boundaries (Marino et al., 2006; Durrani et al., 2013). These n-gram models (also known as tuple sequence model) could help phrase-based translation models to overcome the phrasal independence assumption, but they rely on word alignment to extract bilingual tuples, which brings in additional alignment error (Wu et al., 2014). The other direction lies in utilizing the rule Markov model (Vaswani et al., 2011; Quirk and Menezes, 2006), which directly explores dependencies in rule derivation history and achieves both good performance and slimmer translation model in syntax-based SMT systems. However, the sparsity of translation rules entails aggressive pruning of the training data and constrains the model from scaling to high order grams, significantly limiting the ability of the model.

In this paper we follow the second line and propose a novel recurrent neural network based rule sequence model (RNN-RSM), which utilizes the representational power of recurrent neural network (RNN) to capture arbitrary distance of contextual information in estimating the probability of rule sequences, rather than constrained to n-gram local context limited by Markov assumption. Compared with previous studies, our contributions are as follows:

First, we lift the Markov assumption in rule sequence model and use RNN to capture arbitrary-length of contextual information, which is proven to be more accurate in estimating sequential probabilities (Mikolov et al., 2010).

Second, to alleviate the sparsity of translation rules, we extend our model to factorized RNN-RSM, which incorporates both the source and target side phrase embedding in addition to the translation rule

Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), pages 132–138, Beijing, China, July 26-31, 2015.

history.

Lastly, we apply our model to both phrase-based and hierarchical phrase-based (HPB) systems and achieve an average improvement of 0.9 BLEU points with much slimmer translation models in hypergraph reranking task (Huang, 2008).

2 Rule Sequence Model

We will first brief our rule sequence model with an example from phrase-based system (Koehn et al., 2007). Consider the following translation from Chinese to English:

Bùshí yǔ Shālóng jǔxíng le huìtán
Bush with Sharon hold -ed meeting

'Bush held a meeting with Sharon'

So one possible rule derivation of the above example could be:

$$\dfrac{\dfrac{\dfrac{(0\_\_\_\_\_) : (s_0, \text{“”})}{(\bullet_1\_\_\_\_\_) : (s_1, \text{“Bush”})}\ r_1}{(\bullet\_\_\bullet\bullet\bullet_6) : (s_2, \text{“Bush held talks”})}\ r_2}{(\bullet\bullet\bullet_3\bullet\bullet\bullet) : (s_3, \text{“Bush held talks with Sharon”})}\ r_3$$

r_1: *Bùshí* → Bush
r_2: *jǔxíng le huìtán* → held talks
r_3: *yǔ Shālóng* → with Sharon

Each row is a derivation step, where s_n denotes a hypothesis with a coverage vector capturing the source language words translated so far, and a • in the coverage vector indicates the source word at this position is "covered". Each hypothesis s_{n-1} can be extended into a longer hypothesis s_n by a rule r_n translating an uncovered segment. Note that in phrase-based translation we need to set a distortion limit to prohibit long distance reordering, so the ending position of last phrase is maintained (e.g., $_1$ and $_6$ in the coverage vector).

In our example, translation rules r_1, r_2, r_3 form a *derivation* T which leads to a complete translation. So for rule sequence model, the probability of r_n depends on its derivation history $H(r_n)$:

$$P(r_n) = P(r_n|H(r_n)) \quad (1)$$

and the probability of a rule derivation T is

$$P(T) = \prod_{r_i \in T} P(r_i|H(r_i)) \quad (2)$$

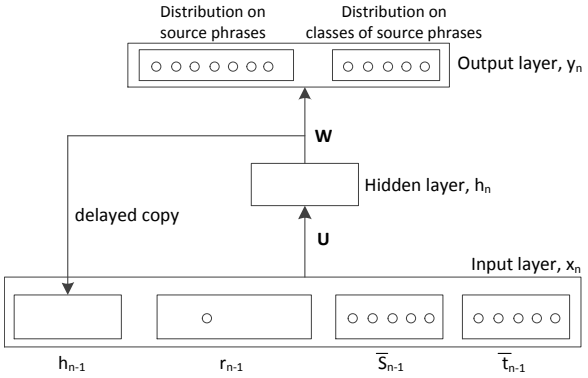

Figure 1: Factorized recurrent neural network with source and target side phrase embeddings.

So the rule sequence model does not make any context independence assumption and generate a rule by looking at a context of previous rules.

2.1 Training

The rule sequence model can then be trained on the path set of rule derivations. To obtain golden derivations of translation rules for each sentence pair, We follow Yu et al. (2013) to utilize force decoding to get golden rule derivations. Specifically, we define a new forced decoding LM which only accepts two consecutive words (denote as p, q) in the reference translation (y_i):

$$P_{forced}(q \mid p) = \begin{cases} 1 & \text{if } \exists j, \text{ s.t. } p = y_j \text{ and } q = y_{j+1} \\ 0 & \text{otherwise} \end{cases}$$

For each hypothesis, we keep the bourndary words as its signiture (only right side for phrase-based model and both sides for HPB). If a boundary word does not occur in the reference, its language model score will be set to $-\infty$; if a boundary word occurs more than once in the reference, the hypothesis is split into multiple hypotheses, one for each index of occurance.

According to the definition, we can see that the rule sequence $[r_1, r_2, r_3]$ in the example could produce the exact reference translation, which is ideal for the training of rule sequence model.

3 Recurrent Neural Network based Rule Sequence Model

In order to capture long-span context, we introduce recurrent neural network based rule sequence model

to estimate the probability $P(r_n|H(r_n))$. Our RNN-RSM can potentially capture arbitrary long context rather than n-1 previous rules limited by Markov assumption. Following Mikolov et al. (2010), we adopt the standard RNN architecture: the input layer encodes previous translation rule using one-hot coding, the output layer produces a probability distribution over all translation rules, and the hidden layer maintains a representation of rule derivation history. However, the standard implementation has severe data sparsity problem due to the large size of rule table couple with the limited training data.

3.1 Factorized RNN-RSM

To solve the sparsity problem, we extend the RNN-RSM model with factorizing rules in the input layer, as shown in Figure 1. It consists of an input layer x, a hidden layer h (state layer), and an output layer y. The connection weights among layers are denoted by matrixes **U** and **W** respectively. Unlike the RNN-RSM, which predicts probability $P(r_n|r_{n-1}, H(r_{n-1}))$, the factorized RNN-RSM predicts probability $P(r_n|r_{n-1}, H(r_{n-1}), \bar{s}_{n-1}, \bar{t}_{n-1})$ to generate following rule r_n, where $\bar{s}_{n-1}$/$\bar{t}_{n-1}$ are the source/target side of r_{n-1}, However, $\bar{s}_{n-1}$ and $\bar{t}_{n-1}$ are still too sparse considering the huge vocabulary size and the diversity in forming phrases, so here we use recursive auto-encoder (Socher et al., 2011; Li et al., 2013) to learn phrase embeddings on both source and target side in an unsupervised mannner, minimizing the reconstruction error.

For those rules that are not contained in the training data, the factorized RNN-RSM backs off to the source/target side embedding $E_{s_{i-1}}$/$E_{t_{i-1}}$. In the special case that $E_{s_{i-1}}$ and $E_{t_{i-1}}$ are dropped, the factorized RNN-RSM goes back to RNN-RSM. Finally, the input layer x_n is formed by concatenating the input vectors and hidden layer h_{n-1} at the preceding time step, as shown in the following equation.

$$x_n = [v^u_{n-1}, v^{\bar{s}}_{n-1}, v^{\bar{t}}_{n-1}, h_{n-1}] \quad (3)$$

The neurons in the hidden and output layers are computed as follows:

$$h_n = f(\mathbf{U} \times x_n), y_n = g(\mathbf{w} \times h_n) \quad (4)$$

$$f(z) = \frac{1}{1+e^{-z}}, g(z) = \frac{e^{z_m}}{\sum_k e^{z_k}} \quad (5)$$

3.2 Factorized RNN-RSM on source and target phrases

The above factorized RNN-RSM is conditioned on the previous context during computing the probability of rule r_n. Since r_n may still suffer from sparsity, we further factorize r_n into its source side phrase $\bar{s}_n$ and target side phrase $\bar{t}_n$. So the probability formula could be rewrite as:

$$\begin{aligned} P(r_n|H(r_n)) &= P(s_n, t_n|H(r_n)) \\ &= P(s_n|H(r_n)) \times P(t_n|s_n, H(r_n)) \end{aligned} \quad (6)$$

The first sub-model $P(s_n, |H(r_n))$ computes the probability distribution over source phrases. Then the second sub-model $P(t_n|s_n, H(r_n))$ computes the probability distribution over t_n that are translated from s_n. The two sub-models are computed with the similar recurrent network shown in Figure 1 except adding the source side information s_n of the current rule r_n into the input layer. This method share the same spirit with the RNN-based translation model (Sundermeyer et al., 2014; Cho et al., 2014), except that we focus on capturing rule dependencies which has a much small search space. Noted that this new factorize model provides richer information for prediction, and actually is faster to train since the vocabulary of source/target phrases are much small than that of the translation rules.

4 Experiments

4.1 Setup

The training corpus consists of 1M sentence pairs with 25M/21M words of Chinese/English respectively. Our development and test set are NIST 2006 and 2008 (newswire portion) respectively.

We obtained alignments by running GIZA++ (Och and Ney, 2004) and used the SRILM toolkit (Stolcke, 2002) to train a 4-gram language model with KN-smoothing on the English side of the training data. Case-insensitive BLEU (Papineni et al., 2002) and MERT (Och, 2003) were used for evaluation and tuning.

We test our method on both phrase-based and hierarchical phrase-based translation models. For phrase-based system, we use Moses with standard features (Koehn et al., 2007). While for hierarchical phrase-based model, we use a in-house implementation of Hiero (Chiang, 2005). We set phrase-limit

System	Moses		Hiero	
	dev-set	test-set	dev-set	test-set
Baseline	28.4	27.7	30.4	30.0
+RMM	28.7	28.3	30.7	30.2
+fRNN-RSM (1)	28.9	**28.6**	30.9	**30.6**
+fRNN-RSM$_{st}$ (2)	**29.3**	**28.5**	**31.2**	**30.7**
+(1)+(2)	**29.6**	**28.7**	**31.4**	**30.8**

Table 1: Main results. RMM is the re-implementation of Vaswani et al. (2011), fRNN-RSM denotes for factorized RNN-RSM describe in Section 3.1, fRNN-RSM$_{st}$ denotes for RNN-RSM factorized by source/target side in Section 3.2. Results in bold mean that the improvements over "Baseline" are statistically significant ($p < 0.05$) (Koehn, 2004).

to 5 for the extraction of both phrase-based rule and SCFG rule, as well as beam size to 100 and distortion limit to 7 in decoding.

Since the rule sequence model belongs to the family of non-local feature (Huang, 2008), traditional testing methods like nbest reranking are not suitable for our experiments. So we adopt *hypergraph reranking* (Huang and Chiang, 2007; Huang, 2008), which proves to be effective for integrating nonlocal features into dynamic programming. The decoding process is divided into two passes. In the first pass, only standard features (i.e., standard features for phrase-based or HPB model) are used to produce a hypergraph. In the second pass, we use the hypergraph reranking algorithm (Huang, 2008) to find promising translations using additional rule sequence feature.

For RNN training, we set the hidden layer size to 512 and classes in the output layer to 256. To obtain phrase-embedding, we use open source tool str2vec[1] (Li et al., 2013) to train two autoencoders on the source and target side of rule-table respectively.

4.2 Results

Table 1 presents the main results of our paper. To show the merits of our RNN-RSM, we also re-implement Vaswani et al. (2011)'s work, denote as rule Markov model (RMM). It utilize tri-gram rule derivation history for prediction, whereas our RNN-RSM could capture arbitrary length of contextual information. We can see that RMM provides a modest improvement over the baseline, 0.6/0.2 points over Moses/Hiero, thanks to the positive guidance of short-span rule dependency. On the other hand, our factorized RNN-RSM with phrase embeddings (fRNN-RSM) provides a more significant BLEU score improvement (0.9 for Moses, 0.6 for Hiero), which exemplifies that the long-span rule dependency captured by RNN could provides additional boost in translation quality. At the same time, factorized RNN-RSM on source and target phrases (fRNN-RSM$_{st}$) alleviate the data sparse problem in RNN training, resulting in slightly better performance. Finally, when we combine both factorized model, we get the best performance at 28.7 for Moses and 30.8 for Hiero, both significantly better than baseline systems.

System	w/o monotone		Full	
	Moses	Hiero	Moses	Hiero
Baseline	27.4	29.8	27.7	30.0
+RMM	27.6	29.9	28.3	30.2
+fRNN-RSM	28.0	30.4	28.6	30.6
+fRNN-RSM$_{st}$	28.2	30.6	28.5	30.7

Table 2: BLEU score comparison on different rule-set, "w/o monotone" denotes we filter out monotone composed rules in both rule table and our RNN-RSM, full denotes we use the total rule-set.

Also, we conduct an interesting experiment to see if our fRNN-RSM could somehow replace the role of composed rules (rules that can be formed out of smaller rules in the grammar) and guides more fine-grained rule-set to produce better translation results. We re-implement He et al. (2009)'s work to filter out monotone composed rules for both Hiero and Moses. We are able to filter out a large number of monotone composed rules, about 50% rules for Hi-

[1] https://github.com/pengli09/str2vec

ero and 31% for Moses. The results are shown in Table 2. Interestingly the performance of slimmer translation model with fRNN-RSM exceeds baseline with full rule-table, and catches up with the original fRNN-RSM. The reason is two-folded: first, deleting monotone composed rules doesn't effect the overall coverage of the rule-set, making limited harm to the system. Second, with less rules, the data sparse problem of RNN training is further alleviated, resulting in a better fRNN-RSM for probability prediction.

5 Related Work

Besides the work of Vaswani et al. (2011) discussed in Section 1, there are several other works using a rule bigram or trigram model in machine translation, Ding and Palmer (2005) use n-gram rule Markov model in the dependency treelet model, Liu and Gildea (2008) applies the same method in a tree-to-string model. Our work is different from theirs in that we lift the Markov assumption and use recurrent neural network to capture much longer contextual information to help probability prediction.

Our work is also in the same spirit with tuple sequence models (Marino et al., 2006; Durrani et al., 2013; Hui Zhang, 2013; Wu et al., 2014), which break the translation sequence into bilingual tuples and use a Markov model to capture the dependency of tuples. Comparing to them, we take a more direct approach to use translation rule dependency to guide translation process, rather than rely on tuples which will be significant affected by word alignment errors.

Outside of machine translation, the idea of weakening independence assumption by modeling the derivation history is also found in parsing (Johnson, 1998), where rule probabilities are conditioned on parent and grand-parent nonterminals. Inspired by it, we successfully find a solution for the translation field.

6 Conclusion

In this paper, we have presented a novel recurrent neural network based rule sequence model to estimate the probability of translation rule sequences. One of the major advantages of our model is its potential to capture long-span dependency compared with n-gram Markov models. In addition, our factorized model with phrase embedding could further alleviate the data sparse problem in RNN training. Finally we conduct experiments on both phrase-based and hierarchical phrase-based models and get an average improvement of 0.9 BLEU points over the baseline. In the future we will investigate stronger network structure such as LSTM to further improve the prediction power of our model.

References

Eugene Charniak, Kevin Knight, and Kenji Yamada. 2003. Syntax-based language models for statistical machine translation. In *Proceedings of MT Summit IX*, pages 40–46. Citeseer.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd ACL*, Ann Arbor, MI.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 541–548. Association for Computational Linguistics.

Nadir Durrani, Alexander Fraser, Helmut Schmid, Hieu Hoang, and Philipp Koehn. 2013. Can markov models over minimal translation units help phrase-based smt? In *ACL (2)*, pages 399–405.

Jesús Giménez and Lluís Màrquez. 2007. Context-aware discriminative phrase selection for statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 159–166. Association for Computational Linguistics.

Zhongjun He, Yao Meng, and Hao Yu. 2009. Discarding monotone composed rule for hierarchical phrase-based statistical machine translation. In *Proceedings of the 3rd International Universal Communication Symposium*, pages 25–29. ACM.

Liang Huang and David Chiang. 2007. Forest rescoring: Fast decoding with integrated language models. In *Proceedings of ACL*, Prague, Czech Rep., June.

Liang Huang. 2008. Forest reranking: Discriminative parsing with non-local features. In *Proceedings of the ACL: HLT*, Columbus, OH, June.

Chris Quirk Jianfeng Gao Hui Zhang, Kristina Toutanova. 2013. Beyond left-to-right: Multiple decomposition structures for smt. In *NAACL*.

Mark Johnson. 1998. Pcfg models of linguistic tree representations. *Computational Linguistics*, 24(4):613–632.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 2007. Moses: open source toolkit for statistical machine translation. In *Proceedings of ACL: Demonstrations*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *EMNLP*, pages 388–395. Citeseer.

Shankar Kumar and William Byrne. 2005. Local phrase reordering models for statistical machine translation. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 161–168. Association for Computational Linguistics.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for itg-based translation. In *EMNLP*, pages 567–577.

Ding Liu and Daniel Gildea. 2008. Improved tree-to-string transducer for machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 62–69. Association for Computational Linguistics.

José B Marino, Rafael E Banchs, Josep M Crego, Adria de Gispert, Patrik Lambert, José AR Fonollosa, and Marta R Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *INTERSPEECH 2010, 11th Annual Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September 26-30, 2010*, pages 1045–1048.

Franz Joseph Och and Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30:417–449.

Franz Joseph Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of ACL*, pages 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL*, pages 311–318, Philadephia, USA, July.

Chris Quirk and Arul Menezes. 2006. Do we need phrases?: challenging the conventional wisdom in statistical machine translation. In *Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics*, pages 9–16. Association for Computational Linguistics.

Libin Shen, Jinxi Xu, and Ralph M Weischedel. 2008. A new string-to-dependency machine translation algorithm with a target dependency language model. In *ACL*, pages 577–585.

Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

Andreas Stolcke. 2002. Srilm - an extensible language modeling toolkit. In *Proceedings of ICSLP*, volume 30, pages 901–904.

Martin Sundermeyer, Tamer Alkhouli, Joern Wuebker, and Hermann Ney. 2014. Translation modeling with bidirectional recurrent neural networks. In *Proceedings of the Conference on Empirical Methods on Natural Language Processing, October*.

Ashish Vaswani, Haitao Mi, Liang Huang, and David Chiang. 2011. Rule markov models for fast tree-to-string translation. In *Proceedings of ACL 2011*, Portland, OR.

Youzheng Wu, Taro Watanabe, and Chiori Hori. 2014. Recurrent neural network-based tuple sequence model for machine translation. In *Proc. COLING*, pages 1908–1917.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 521–528. Association for Computational Linguistics.

Heng Yu, Liang Huang, Haitao Mi, and Kai Zhao. 2013. Max-violation perceptron and forced decoding for scalable mt training. In *EMNLP*, pages 1112–1123.

Heng Yu, Haitao Mi, Liang Huang, and Qun Liu. 2014. A structured language model for incremental tree-to-string translation.

Richard Zens and Hermann Ney. 2006. Discriminative reordering models for statistical machine translation. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 55–63. Association for Computational Linguistics.