Exploring Recombination for Efficient Decoding of Neural Machine Translation

Zhisong Zhang[*], **Rui Wang**[2,†], **Masao Utiyama**[2], **Eiichiro Sumita**[2] **and Hai Zhao**[1,†]
[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2]National Institute of Information and Communications Technology (NICT)
zerozones17@gmail.com,
{wangrui, mutiyama, eiichiro.sumita}@nict.go.jp,
zhaohai@cs.sjtu.edu.cn

Abstract

In Neural Machine Translation (NMT), the decoder can capture the features of the entire prediction history with neural connections and representations. This means that partial hypotheses with different prefixes will be regarded differently no matter how similar they are. However, this might be inefficient since some partial hypotheses can contain only local differences that will not influence future predictions. In this work, we introduce recombination in NMT decoding based on the concept of the "equivalence" of partial hypotheses. Heuristically, we use a simple n-gram suffix based equivalence function and adapt it into beam search decoding. Through experiments on large-scale Chinese-to-English and English-to-Germen translation tasks, we show that the proposed method can obtain similar translation quality with a smaller beam size, making NMT decoding more efficient.

1 Introduction

Recently, end-to-end Neural Machine Translation (NMT) models (Sutskever et al., 2014; Bahdanau et al., 2015) have achieved notable success. A remarkable characteristic of NMT is that the decoder, which is typically implemented using Recurrent Neural Network (RNN), can capture the features of the entire decoding history. This model does not depend on any independence assumptions and treats sequences with different prefixes as totally different hypotheses. However, many of the NMT output sequences are quite similar and they typically contain only local differences that do not influence future decoding significantly.

Src	有 消息 说 , 这 两 个 城市 的 工人 已经 成立 了 独立 工会 .
Ref	some sources said that the workers in these two *cities* have established ...
Output	according to some sources , the workers of these two *cities* have(-0.075) already(-0.331) set(-0.536) up(-0.001) ...
	according to some sources , workers of these two *cities* have(-0.073) already(-0.248) set(-0.783) up(-0.001) ...
	it has been reported that the workers of these two *cities* have(-0.058) already(-0.414) set(-0.608) up(-0.001) ...

Table 1: Example of similar partial hypotheses in beam search. The hidden layers of the partial hypotheses ending with "*cities*" correspond to the nodes boxed in Figure 1 (only three hypotheses are listed for brevity). The negative log probabilities calculated by the model for the words predicted after "*cities*" are given in parentheses.

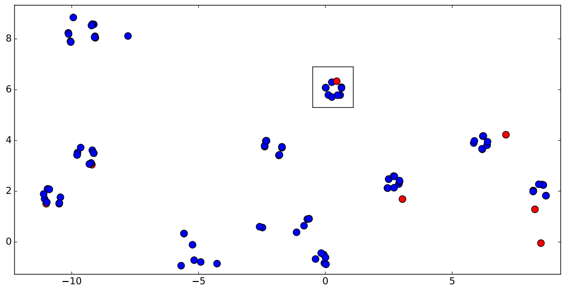

Figure 1: t-SNE visualization (Maaten and Hinton, 2008) of the recurrent hidden layer vectors for partial hypotheses for the example in Table 1. Reference and prediction hypotheses are presented as red and blue nodes, respectively. The nodes inside the box represent the hidden features of partial hypotheses ending with "*cities*".

Table 1 and Figure 1 present an example of such pattern of local differences in NMT decoding. As shown in Table 1, the three partial hypotheses that

[*]Zhisong Zhang was a graduate student at SJTU and a research intern at NICT when conducting this work. This work is partially supported by the program "Promotion of Global Communications Plan: Research, Development, and Social Demonstration of Multilingual Speech Translation Technology" of MIC, Japan. Hai Zhao was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04). Thanks a lot for the helpful discussions with Kehai Chen.

[†]Co-corresponding authors

Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, pages 4785–4790
Brussels, Belgium, October 31 - November 4, 2018. ©2018 Association for Computational Linguistics

Algorithm 1 Merging for Beam Search.

```
Require: list of sorted candidates $C$; beam size $k$;
    equivalence function $Eq$.
Ensure: list of candidates surviving in the beam: $C'$.
 1: $C'$ = [ ]
 2: # Scan according to the sorted order.
 3: for $c$ in $C$:
 4:     merge_flag = False
 5:     # Check previous surviving states for merging.
 6:     for $s$ in $C'$:
 7:         # Check with candidate merger states.
 8:         for $s'$ in sequence($s$):
 9:             if $Eq(c, s')$ and score($c$)<score($s'$):
10:                 merge_flag = True
11:     # Pruning by the merger.
12:     if not merge_flag:
13:         $C'$.append($c$)
14:     # Pruning by the beam size.
15:     if $len(C') >= k$:
16:         break
17: return $C'$
```

end with "*cities*" share similar patterns. Firstly, as shown in Figure 1, their hidden layer features are close in the latent space. Moreover, for future predictions, the model predicts identical sequences and gives similar scores for them. Although going through different paths, these partial hypotheses appear to be similar or likely equivalent.

Intuitively, for efficiency, we do not need to expand all of these partial hypotheses (states) since they have similar future predictions. In fact, this corresponds to the idea of hypothesis recombination (also known as state merging, which will be used interchangeably) from traditional Phrase-Based Statistical Machine Translation (PBSMT) (Koehn et al., 2003). Given a method to find mergeable states, we can employ recombination in NMT decoding as well.

In this paper, we adopt the mechanism of recombination in NMT decoding based on the definition of "equivalence" of partial hypotheses. Heuristically, we try a simple n-gram suffix based equivalence function and apply it to beam search without adding any neural computation cost. Through experiments on two large-scale translation tasks, we show that it can help to make the decoding more efficient.

Most recent NMT studies have focused on model improvement (Luong et al., 2015; Tu et al., 2016b; Gehring et al., 2017; Vaswani et al., 2017), and only a few have studied the search problem directly. For example, Khayrallah et al. (2017) and Stahlberg et al. (2016) explored searching on lattices generated by traditional Statistical Machine Translation (SMT). In addition, Freitag and Al-Onaizan (2017) investigated different beam search pruning strategies; however, they primarily focused on pruning candidates locally. (Niehues et al., 2017) analyzed the effects of modeling and searching, but focused on re-ranking analysis. Rather than considering candidates from other model's k-best lists, we focus on the own exploration space of a single NMT model and provide a method for more efficient searching.

2 Method

For state merging, "equivalence" should be defined from the aspect of future predictions: states with the same predictions in the future decoding process can be regarded as equivalent. We use an equivalence function $Eq(s_1, s_2)$ to denote that the two states s_1 and s_2 can be regarded as equivalent.

With the concept of equivalence, we can build the method of recombination over it. There are mainly two problems to solve:

1. How to merge states given function Eq? (§2.1)
2. How to obtain this equivalence function? (§2.2)

2.1 Search with Merging

To adopt an equivalence function $Eq(s_1, s_2)$ to merge states in a search process, we need to specify the logic of the merging mechanism. Here, without loss of generality, we specifically focus on the typical beam search.

We adopt merging in NMT beam search with a simple method: retaining the word-level search process and adding a state merger when pruning the beam at each time step. Algorithm 1 shows the proposed merging-enhanced pruning method.

Ordinary beam search only prunes candidates based on beam size (Lines 15-16), while the proposed method adds a merger to prune extra equivalent states (Lines 6-10). To manage the merging process, candidate list C are ordered[1] by model score and considered in turn. When checking equivalence for one candidate state c, we consider all current-step surviving states and their previous-step antecedences. We include previous-step states, because equivalent states may have different sequence lengths and thus not be in the same beam-search step. In Line 8, we define "sequence" as a function of obtaining the possible states that

[1] In plain beam search, the candidates may not need to be sorted. We use a local selector to make the sorting efficient: a local k-best selector is first applied on each previous-step candidate states, making the size of the candidate list at most $k * k$ rather than $k * |V|$, where $|V|$ is the vocabulary size.

can merge the current candidate c. If a candidate state c is not merged with any higher-ranked state, it is added to the surviving list C' (Line 13) and can possibly merge the lower-ranked ones later.

When deciding whether to merge, we also consider a criterion on model scores: we only merge state c when its score is lower than s'. Since we also consider previous-step states with different sequence lengths, a length reward λ is added for this comparison of partial hypotheses: $\text{score}(s) = \sum_{y \in s} \lambda + \log \text{p}(y)$. We also attempted length normalization, but found it performed slightly worse.

The merged partial hypotheses can be stored, and by assuming that their future predictions will be the same as their mergers, a lattice-like translation graph can be obtained. We can further extract k-best list from this structure using another beam-search on the lattice (also with length reward when comparing partial hypotheses). Note that this beam search process can be fast, since we reuse the model scores from previous search and no extra neural computations will be included.

2.2 Equivalence Function

Finding an exact equivalence function for NMT is difficult, because future predictions relies on the features from the entire previous sequence and any different sequences are not the same according to the NMT model. Here, we consider a n-gram suffix based heuristic approximation for this problem.

We adopt an approximate equivalence function:

$$\begin{aligned} Eq'(s_1, s_2) \;\equiv\; & s_1.suffix(n) = s_2.suffix(n) \\ & \wedge |s_1.length - s_2.length| < r \end{aligned}$$

Here, $suffix(n)$ represents the n-gram suffix of the sequence of a state, and r is the threshold for the length different of the two states.

This definition of equivalence only considers a subset of state features, which are inspired by PBSMT. In PBSMT, different sequences could lead to states with identical features based on n-gram suffix, and these states are exactly equivalent. Although this is not the case for NMT, the subset may encodes important and relevant features.

Although this function is simple and brings extra approximation, it has the merit of efficiency. In Algorithm 1, we can store the n-gram features of the surviving states in a hash-map and replace the for-loop checking (Line 6-10) with hashing, making the extra time-complexity O(1) for each state. During experiments, we found the extra cost brought by feature matching is far less than the cost of original neural computation.

3 Experiments and Analysis

The proposed method was evaluated on two translation tasks: NIST Chinese-English (Zh-En) and WMT English-German (En-De). For Zh-En, the training set comprised 1.4M sentences pairs from LDC corpora. NIST 02 was selected as the development set and NIST 03 to 06 were used for testing. For En-De, 4.5M WMT training data were utilized, the concatenation of newstest 2012 and 2013 was adopted as the development set, and newstest 2014 to 2016 were adopted as the test set.

We implemented[2] an attentional RNN-based NMT model and its decoder in Python with the DyNet toolkit (Neubig et al., 2017). All the experiments were carried out on one P100 GPU. For Zh-En, we set the vocabulary size of both sides to 30K, and for En-De, we adopted 50K BPE operations (Sennrich et al., 2016). The evaluation metric was tokenized BLEU (Papineni et al., 2002) calculated by `multi-bleu.perl`. Detailed settings can be found in the supplementary material.

We added a local threshold pruner to exclude unlikely words whose probabilities were less than 10% of the highest and adopted length normalization for final hypotheses ranking. For comparing partial hypotheses, the length reward λ was set to 1.0 and 0.4 for Zh-En and En-De, respectively. For the equivalence function, we utilized a suffix of 4-gram and a length difference threshold r of 2.

These hyper-parameters were set by preliminary experiments. For the length difference threshold r, we found that relatively small r like 1 or 2 was better than larger ones, which is reasonable since if the merged hypotheses differs too much in length, there are higher chances that they covered different information. For n-gram suffix, we found smaller n-grams made more bad merges and 4-gram is a reasonably good choice, slightly larger ones gave slightly worse results and also less chances of recombination.

3.1 Results

Figure 2 show the results of various beam sizes on the concatenation of all test sets. Separate results are given in the supplementary material.

As shown by the speed curves, merging adds little extra cost (less than 10%) to decoding at

[2] https://github.com/zzsfornlp/znmt-merge

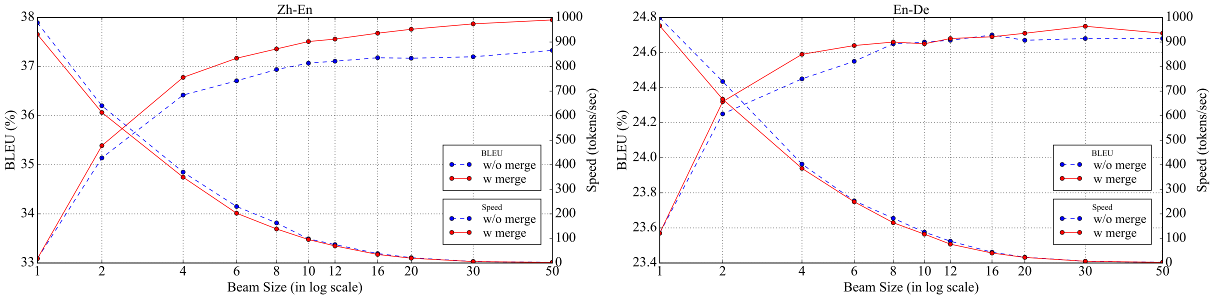

Figure 2: Translation quality and speed of Zh-En and En-De test sets (5453 sentences by concatenating NIST 03 to 06 and 8171 sentences by concatenating newstest 2014 to 2016, respectively).

the same beam size. Moreover, since bringing no extra neural computations, the proposed merging mechanism is transparent to neural architectures and easy to adopt. In our experiments, we used batched decoding on GPU and merging did not influence the efficiency of this implementation.

For translation quality, the results indicate that the proposed methods can yield improvements at various beam sizes for Zh-En and small beam sizes for En-De. Moreover, in some way, merging can make the search more efficient. For example, in both datasets, merge-enhanced searchers with beam-size 6 can obtain comparable or better results compared to those of ordinary searchers with beam-size 12 (on BLEU, 37.17 vs. 37.11 for Zh-En, 24.64 vs. 24.67 for En-De). As for decoding speed, the one of beam-size 6 can be more than twice of the one of beam-size 12 (over 200 tokens/second vs. around 100 tokens/second). That is to say, with merging, we can achieve similar translation quality with a smaller beam size, which leads to higher decoding speed.

The results show that for large beam sizes, expanding explored search space by increasing beam size or adopting merging helps more in Zh-En than En-De. A possible explanation for this is that in NIST Zh-En dataset, each source sentences has four references for evaluation, which encourages the diversity brought by expanding reached search space. In Table 2, we compare the BLEU scores with multiple and single references on several beam sizes, and the single-reference results does not always increase along the beam size like the multiple ones. The En-De dataset also has only one reference and is similar to this case.

The results also show that expanding explored search space does not always bring improvements. This concerns more on modeling than searching and corresponds with previous findings on the relations between NMT searching and modeling (Tu et al., 2016a; Niehues et al., 2017; Li et al., 2018). The potential of the proposed method might be better realized with improved NMT models.

Ref \ Beam	10	16	30	50
Multi-Ref	37.07	37.17	37.19	37.32
Single-Ref	18.23	18.29	18.23	18.26

Table 2: Comparisons of multi- and single-reference BLEU scores of NIST 03-06 with “w/o merging”. “Multi-Ref” uses all four references, and “Single-Ref” only uses the second one, whose evaluations disagree most with “Multi-ref”.

3.2 Analysis

We further analyzed the merge-enhanced search process. For these analyses, we mainly checked decoding with a beam size of 10 on Zh-En dataset.

Frequency of Merging First, we investigated how often recombination occurs and how much it expands the explored output space. For a beam size of 10, with influences from the local pruner and the proposed merger, the average expanding size is 7.60 for each step, and the average number of merger-pruned partial hypotheses is 0.61 per step (22.5 per sentence). This indicates that a partial hypothesis is recombined in every two steps. The output translation graph can hold much more output space than the original k-best list, and we found that on average the possible output sequences were averagely 200 times the beam size. Figure 3 shows an example of the output translation graph.

Merging and Similarity of Hidden States It is nearly impossible to explore such a large space with an exact NMT model; thus, we depend on the assumption that merged hypotheses have nearly the same features. To evaluate this assumption, we calculated the similarity between the hidden layers of the merged partial hypotheses. Among the 122772 merge points in 5453 Zh-En sentences, the average cosine similarity (in range $[-1, 1]$) was 0.986, which indicates that the recombinations are reasonable. In addition, we tried adding simple cosine similarity constraints (using another

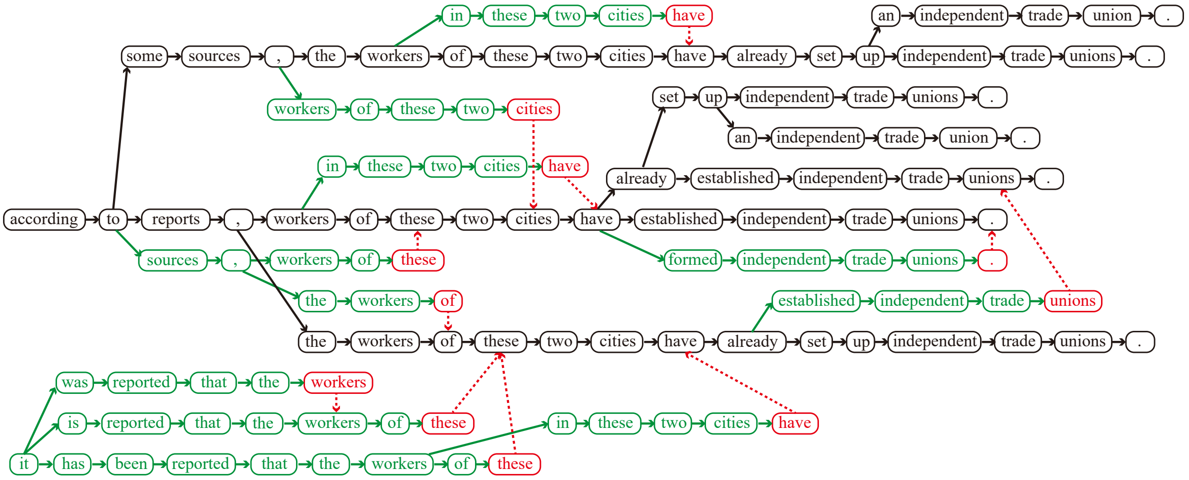
in these two cities have
an independent trade union .
some sources , the workers of these two cities have already set up independent trade unions .
set up independent trade unions .
workers of these two cities
an independent trade union .
in these two cities have
already established independent trade unions .
according to reports , workers of these two cities have established independent trade unions .
sources , workers of these
formed independent trade unions .
the workers of
established independent trade unions
the workers of these two cities have already set up independent trade unions .
was reported that the workers
is reported that the workers of these
in these two cities have
it has been reported that the workers of these

Figure 3: An example output translation graph. The red nodes and dashed arrows indicate merge points.

Beam	w/ merge	w/o merge
1	1.6% / 90.6%	1.6% / 90.6%
4	8.0% / 51.3%	5.2% / 54.8%
10	29.1% / 15.7%	–
16	44.5% / 11.8%	28.2% / 5.6%
30	64.6% / 20.3%	56.4% / 20.2%

Table 3: Comparisons of prediction model scores between different searching settings and a basic setting, which is "Beam=10, w/o merge". The pattern "$a\%$ / $b\%$" means that compared with the basic setting, $a\%$ of the sentences get higher model scores and $b\%$ get lower ones. For the rest (1-$a\%$-$b\%$), they give identical predictions.

threshold) in the equivalence function, however, we found that this does not bring obvious additional benefits.

Effects of Merging We further conducted comparisons between the predictions of ordinary and merge-enhanced beam search. First, we investigated the model scores of their predictions. As shown in Table 3, we selected "Beam=10, no merge" as the basic setting, and compared the predictions of other settings with it. Overall, the merge-enhanced searcher can obtain higher model score predictions, which suggests its stronger search ability, because the goal of searching is to return hypotheses with higher model scores.

Moreover, we tried a re-ranking experiment on 100-best lists with 4-checkpoint-model-ensemble, and only found similar slight improvements for plain and merge-enhanced search. Nevertheless, since merge-enhanced search can obtain a output translation graph, we expect that the graph can contain more diverse hypotheses.

To verify this, we compared the oracle BLEU scores within the reached space. To extract oracle hypotheses from the translation graphs, we simply adopted approximate Partial BLEU Oracle (Dreyer et al., 2007; Sokolov et al., 2012). Merge-based searcher could obtain an oracle score of 47.83, while ordinary beam searcher could only get 42.57. Only by increasing the beam size up to 100 could the ordinary beam searcher achieve a better result of 48.74. This indicates that recombination helps to touch more output space.

4 Conclusion and Discussion

In this work, 1) we show that decoding with heuristic recombination can obtain similar translation qualities with smaller beam sizes, thus increasing efficiency, and, 2) we empirically explore the decoding process and analyze the influences of recombination from various aspects.

Although the improvements brought by recombination depend on careful refinements of the model, this concerns more on modeling, since the goal of decoding is to find hypotheses with higher model scores. The potential of recombination may be further realized by improving how the output sequences are modeled. Another interesting topic will be the combination with SMT or extra larger language models (Wang et al., 2013, 2014).

For the equivalence function, there can also be extensions. For example, a model-based equivalence function can be trained by using the neural features (hidden layers in RNN). However, model-based equivalence functions may bring extra neural computation cost and be harder to efficiently implemented. In this work, we focus on the merging mechanism and leave the study of equivalence function for future work.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *ICLR*.

Markus Dreyer, Keith Hall, and Sanjeev Khudanpur. 2007. Comparing reordering constraints for smt using efficient bleu oracle computation. In *Proceedings of SSST, NAACL-HLT 2007 / AMTA Workshop on Syntax and Structure in Statistical Translation*, pages 103–110, Rochester, New York.

Markus Freitag and Yaser Al-Onaizan. 2017. Beam search strategies for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 56–60, Vancouver.

Jonas Gehring, Michael Auli, David Grangier, and Yann Dauphin. 2017. A convolutional encoder model for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 123–135, Vancouver, Canada.

Huda Khayrallah, Gaurav Kumar, Kevin Duh, Matt Post, and Philipp Koehn. 2017. Neural lattice search for domain adaptation in machine translation. In *Proceedings of IJCNLP*, pages 20–25, Taipei, Taiwan.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54.

Yanyang Li, Tong Xiao, Yinqiao Li, Qiang Wang, Changming Xu, and Jingbo Zhu. 2018. A simple and effective approach to coverage-aware neural machine translation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 292–297.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of machine learning research*, 9(Nov):2579–2605.

Graham Neubig, Chris Dyer, Yoav Goldberg, Austin Matthews, Waleed Ammar, Antonios Anastasopoulos, Miguel Ballesteros, David Chiang, Daniel Clothiaux, Trevor Cohn, et al. 2017. Dynet: The dynamic neural network toolkit. *arXiv preprint arXiv:1701.03980*.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2017. Analyzing neural mt search and model performance. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 11–17, Vancouver.

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.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1715–1725, Berlin, Germany.

Artem Sokolov, Guillaume Wisniewski, and Francois Yvon. 2012. Computing lattice bleu oracle scores for machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 120–129, Avignon, France.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016. Syntactically guided neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 299–305, Berlin, Germany.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*, pages 3104–3112.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2016a. Neural machine translation with reconstruction. In *Proceedings of AAAI*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016b. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 76–85, Berlin, Germany.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 6000–6010.

Rui Wang, Masao Utiyama, Isao Goto, Eiichro Sumita, Hai Zhao, and Bao-Liang Lu. 2013. Converting continuous-space language models into n-gram language models for statistical machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, Seattle, Washington, USA.

Rui Wang, Hai Zhao, Bao-Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2014. Neural network based bilingual language model growing for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar.