Supplementary Material of "Exploring Recombination for Efficient Decoding of Neural Machine Translation"

1 Settings

For the NMT model, we followed the relatively standard practice of hyper-parameter settings. Both the encoder and decoder adopted one-layer Recurrent Neural Network with Gate Recurrent Units (Cho et al., 2014). The encoder obtained the representations of source words by concatenating the hidden layer from bidirectional RNNs. The decoder initialized its hidden layer with the average of source representations and adopted conditional GRU with attention (Sennrich et al., 2017). Embeddings had the dimensions of 512, and the sizes of hidden layers of the RNNs were set to 1000.

For model training, we adopted the loss function of word-level cross-entropy. We discarded sentences that are longer than 50 words in training and trained the model with mini-batches of 80. Adam (Kingma and Ba, 2014) with an initial learning rate of 0.0001 was utilized for optimization. The models were trained for 30 and 10 epochs for Zh-En and En-De, respectively. During training, we also adopted dropouts of 0.2 for embeddings, recurrent layers and output hidden layers.

For Zh-En, we tokenized the Chinese side with Stanford Word Segmenter (Chang et al., 2008) and lowercased the English words. The vocabulary sizes were set to 30K and out-of-vocabulary words were replaced by a special token. For En-De, we adopted 50K joint Byte Pair Encoding (BPE) operations (Sennrich et al., 2016) for the dataset after true-casing. For evaluations, following previous conventions, we used case-insensitive and case-sensitive tokenized BLEU scores for Zh-En and En-De, respectively.

2 Evaluations

Tables 1 and 2 show the detailed separate evaluations on test sets with beam sizes of 6 and 12. Although the BLEU differences are relatively small, we still find steady improvements from merge-enhanced searcher in some datasets. This might be due to the fact that only the search process is changed, while the underlying models are the same.

Beam	Strategy	Avg. Speed	NIST-03	NIST-04	NIST-05	NIST-06
6	w/o merge	230.13	36.72	39.59	35.57	33.91
	w/ merge	212.70	37.10*	40.03**	36.06**	34.41**
12	w/o merge	74.89	37.20	39.77	36.11	34.42
	w/ merge	69.56	37.53	40.20**	36.66**	34.87**

Table 1: Evaluations of separate test sets of Zh-EN with beam sizes of 6 and 12. "Avg. Speed" denotes averaged decoding speed in tokens per second. "*" indicates merge-enhanced decoder is statistically significantly better (Koehn, 2004) than the one without merging at $p < 0.05$, and "**" indicates $p < 0.01$.

Beam	Strategy	Avg. Speed	newstest2014	newstest2015	newstest2016
6	w/o merge	253.20	21.04	23.88	28.59
	w/ merge	249.25	21.12	24.03	28.65
12	w/o merge	88.78	21.06	24.11	28.72
	w/ merge	78.73	21.22*	24.04	28.65

Table 2: Evaluations of separate test sets of En-De. (Using the same notations as Table 1)

3 Output Example

Here, taking the exemplary sentence pair (also shown in Table 3) in the main content as the instance, we show the outputs of ordinary beam search (k-best list) and merge-enhanced search (translation graph). We adopt a beam size of 10 for both searchers.

Source	有 消息 说 , 这 两 个 城市 的 工人 已经 成立 了 独立 工会 .
Reference	some sources said that the workers in these two cities have established an independent labor union .

Table 3: Example translation pair.

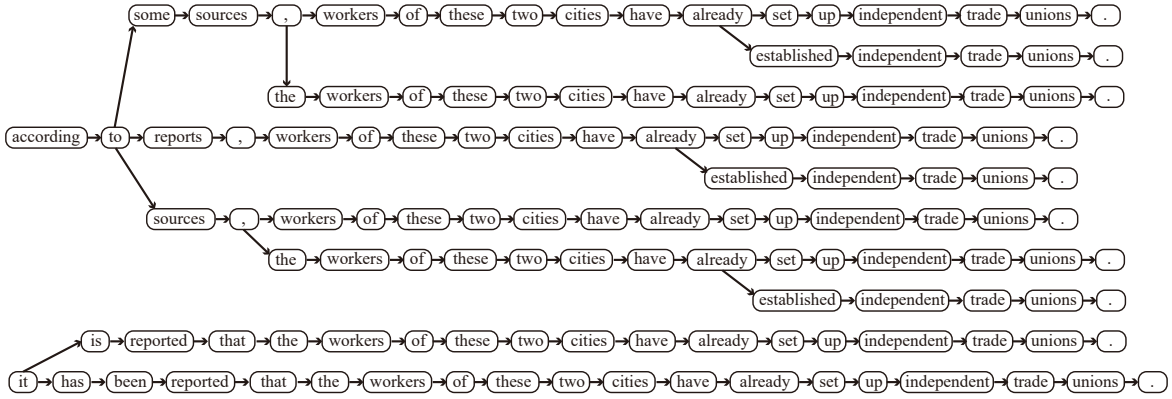

Figure 1: The k-best outputs of ordinary beam search with a beam size of 10.

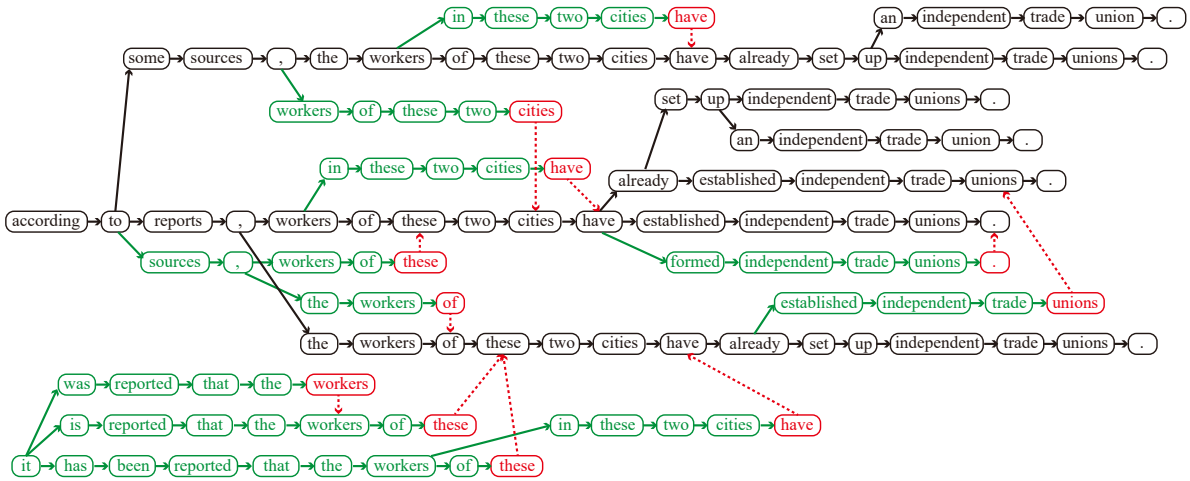

Figure 2: The output translation graph of merge-enhanced beam search with a beam size of 10. Here, black paths indicate the partial hypotheses that are not merged and can reach the final end state, and green paths indicate the partial hypotheses that are merged according to a 4-gram matching criterion, while the red nodes and dashed arrows indicate at which word and with which state they are merged. For simplicity, we only show 11 out of the 22 merged paths.

Figure 1 presents the k-best list of ordinary beam search, and Figure 2 shows the translation graph obtained through merge-enhanced beam search. We can see that the translation graph can hold more possible translations than k-best list; for example, in the translation graph, there are several paths encoding the phrase "in these two cities" of the reference, which does not appear in the k-best list. Moreover, many candidates in the k-best list have only local differences, and can be further compacted into a lattice-like structure. This actually corresponds to our motivation for the introduction of recombination in NMT.

References

Pi-Chuan Chang, Michel Galley, and Christopher D. Manning. 2008. Optimizing Chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 224–232, Columbus, Ohio.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734, Doha, Qatar.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of EMNLP*, pages 388–395, Barcelona, Spain.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017. Nematus: a toolkit for neural machine translation. In *Proceedings of the Software Demonstrations of EACL*, pages 65–68, Valencia, Spain.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725, Berlin, Germany.