A Simple and Effective Approach to Coverage-Aware Neural Machine Translation

Yanyang Li[1], **Tong Xiao**[1], **Yinqiao Li**[1], **Qiang Wang**[1], **Changming Xu**[1] and **Xueqiang Lu**[2]
[1]Natural Language Processing Lab., Northeastern University
[2]Beijing Key Laboratory of Internet Culture and Digital Dissemination Research
blamedrlee@outlook.com, xiaotong@mail.neu.edu.cn, li.yin.qiao.2012@hotmail.com, wangqiangneu@gmail.com, changmingxu@neuq.edu.cn, lxq@bistu.edu.cn

Abstract

We offer a simple and effective method to seek a better balance between model confidence and length preference for Neural Machine Translation (NMT). Unlike the popular length normalization and coverage models, our model does not require training nor reranking the limited n-best outputs. Moreover, it is robust to large beam sizes, which is not well studied in previous work. On the Chinese-English and English-German translation tasks, our approach yields +0.4 $\sim$ 1.5 BLEU improvements over the state-of-the-art baselines.

1 Introduction

In the past few years, Neural Machine Translation (NMT) has achieved state-of-the-art performance in many translation tasks. It models the translation problem using neural networks with no assumption of the hidden structures between two languages, and learns the model parameters from bilingual texts in an end-to-end fashion (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Cho et al., 2014). In such systems, target words are generated over a sequence of time steps. The model score is simply defined as the sum of the log-scale word probabilities:

$$\log \mathrm{P}(\mathbf{y}|\mathbf{x}) = \sum_{j=1}^{|\mathbf{y}|} \log \mathrm{P}(y_j|y_{<j}, \mathbf{x}) \qquad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the source and target sentences, and $\mathrm{P}(y_j|y_{<j}, \mathbf{x})$ is the probability of generating the j-th word y_j given the previously-generated words $y_{<j}$ and the source sentence $\mathbf{x}$.

However, the straightforward implementation of this model suffers from many problems, the most obvious one being the bias that the system tends to choose shorter translations because the log-probability is added over time steps. The situation is worse when we use beam search where the shorter translations have more chances to beat the longer ones. It is in general to normalize the model score by translation length (say length normalization) to eliminate this system bias (Wu et al., 2016).

Though widely used, length normalization is not a perfect solution. NMT systems still have under-translation and over-translation problem even with a normalized model. It is due to the lack of the *coverage model* that indicates the degree a source word is translated. As an extreme case, a source word might be translated for several times, which results in many duplicated target words. Several research groups have proposed solutions to this bad case (Tu et al., 2016; Mi et al., 2016). E.g., Tu et al. (2016) developed a coverage-based model to measure the fractional count that a source word is translated during decoding. It can be jointly learned with the NMT model. Alternatively, one can rerank the n-best outputs by coverage-sensitive models, but this method just affects the final output list which has a very limited scope (Wu et al., 2016).

In this paper we present a simple and effective approach by introducing a coverage-based feature into NMT. Unlike previous studies, we do not resort to developing extra models nor reranking the limited n-best translations. Instead, we develop a coverage score and apply it to each decoding step. Our approach has several benefits,

- Our approach does not require to train a huge neural network and is easy to implement.
- Our approach works on beam search for each target position and thus can access more translation hypotheses.
- Our approach works consistently well under different sized beam search and sentence lengths contrary to what is observed in other systems (Koehn and Knowles, 2017).

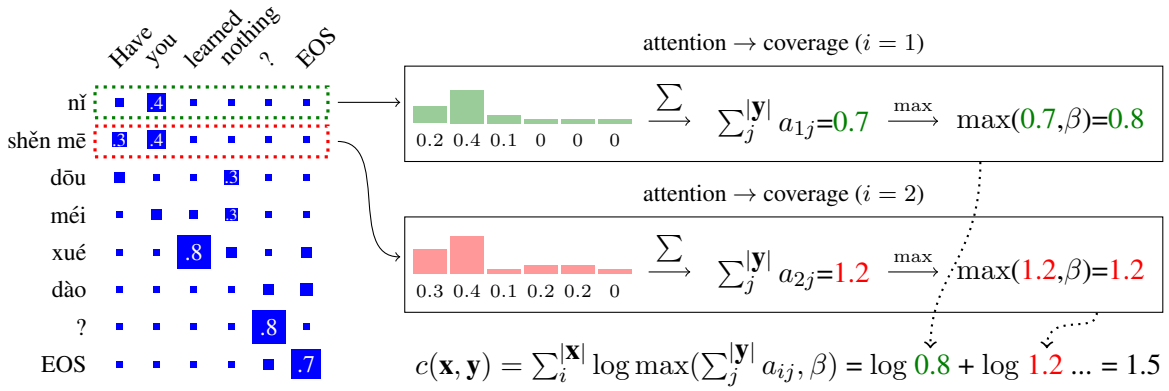

Figure 1: The coverage score for a running example (Chinese pinyin-English and $\beta = 0.8$).

We test our approach on the NIST Chinese-English and WMT English-German translation tasks, and it outperforms several state-of-the-art baselines by 0.4∼1.5 BLEU points.

2 The Coverage Score

Given a word sequence, a coverage vector indicates whether the word of each position is translated. This is trivial for statistical machine translation (Koehn, 2009) because there is no overlap between the translation units of a hypothesis, i.e., we have a 0-1 coverage vector.

However, it is not the case for NMT where the coverage is modeled in a soft way. In NMT, no explicit translation units or rules are used. The attention mechanism is used instead to model the correspondence between a source position and a target position (Bahdanau et al., 2015). For a given target position j, the attention-based NMT computes attention score a_{ij} for each source position i. a_{ij} can be regarded as the measure of the correspondent strength between i and j, and is normalized over all source positions (i.e., $\sum_i^{|\mathbf{x}|} a_{ij} = 1$) [1].

Here, we present a coverage score (CS) to describe to what extent the source words are translated. In principle, the coverage score should be high if the translation covers most words in source sentence, and low if it covers only a few of them. Given a source position i, we define its *coverage* as the sum of the past attention probabilities $c_i = \sum_j^{|\mathbf{y}|} a_{ij}$ (Wu et al., 2016; Tu et al., 2016). Then, the coverage score of the sentence pair $(\mathbf{x}, \mathbf{y})$ is defined as the sum of the truncated coverage over all positions (See Figure 1 for an illustration):

$$c(\mathbf{x},\mathbf{y}) = \sum_i^{|\mathbf{x}|} \log \max(\sum_j^{|\mathbf{y}|} a_{ij}, \beta) \quad (2)$$

where β is a parameter that can be tuned on a development set. This model has two properties:

- **Non-linearity** Eq. (2) is a log-linear model. It is desirable because this model does not benefit too much from the received attention when the coverage of a source word is high. This can prevent the cases that the system puts too much attention on a few words while others only receive a little attention to have relatively high scores. Beyond this, the log-scale scoring fits into the NMT model where word probabilities are represented in the logarithm manner (See Eq. (1)).

- **Truncation** At the early stage of decoding, the coverage of the most source words is close to 0. This may result in a negative infinity value after the logarithm function, and discard hypotheses with sharp attention distributions, which is not necessarily bad. The truncation with the lowest value β can ensure that the coverage score has a reasonable value. Here β is similar to model warm-up, which makes the model easy to run in the first few decoding steps. Note that our way of truncation is different from Wu et al. (2016)'s, where they clip the coverage into [0, 1] and ignore the fact that a source word may be translated into multiple target words and its coverage should be of a value larger than 1.

For decoding, we incorporate the coverage score into beam search via linear combination with the NMT model score as below,

[1] As the discussion of the attention mechanism is out of the scope of this work, we refer the reader to Bahdanau et al. (2015); Luong et al. (2015) for more details.

$$s(\mathbf{x}, \mathbf{y}) = (1 - \alpha) \cdot \log \mathrm{P}(\mathbf{y}|\mathbf{x}) + \alpha \cdot c(\mathbf{x}, \mathbf{y}) \quad (3)$$

where $\mathbf{y}$ is a partial translation generated during decoding, $\log \mathrm{P}(\mathbf{y}|\mathbf{x})$ is the model score, and α is the coefficient for linear interpolation.

In standard implementation of NMT systems, once a hypothesis is finished, it is removed from the beam and the beam shrinks accordingly. Here we choose a different decoding strategy. We keep the finished hypotheses in the beam until the decoding completes, which means that we compare the finished hypotheses with partial translations at each step. This method helps because it can dynamically determine whether a finished hypothesis is kept in beam through the entire decoding process, and thus reduce search errors. It enables the decoder to throw away finished hypotheses if they have very low coverage but are of high likelihood values.

3 Experiments

3.1 Setup

We evaluated our approach on Chinese-English and German-English translation tasks. We used 1.8M sentence Chinese-English bitext provided within NIST12 OpenMT[2] and 4.5M sentence German-English bitext provided within WMT16. For Chinese-English translation, we chose the evaluation data of NIST MT06 as the development set, and MT08 as the test set. All Chinese sentences were word segmented using the tool provided within NiuTrans (Xiao et al., 2012). For German-English translation, we chose newstest2013 as the development set and newstest2014 as the test set.

Our baseline systems were based on the open-source implementation of the NMT model presented in Luong et al. (2017). The model was consisted of a 4-layer bi-directional LSTM encoder and a 4-layer LSTM decoder. The size of the embedding and hidden layers was set to 1024. We applied the additive attention model on top of the multi-layer LSTMs (Bahdanau et al., 2015). For training, we used the Adam optimizer (Kingma and Ba, 2015) where the learning rate and batch size were set to 0.001 and 128. We selected the top 30k entries for both source and target vocabularies. For the English-German task, BPE (Sennrich et al., 2016) was used for better performance.

[2]LDC2000T46, LDC2000T47, LDC2000T50, LDC2003E14, LDC2005T10, LDC2002E18, LDC2007T09, LDC2004T08

	Entry	Zh-En		En-De	
		dev	test	dev	test
b=10	base	37.55	30.91	23.72	23.36
	LN	38.85	32.32	23.96	22.93
	CP	38.68	31.84	23.92	23.27
	CP†	35.93	29.98	23.67	23.53
	LN+CP	39.07	32.47	23.98	23.26
	CS	39.13	32.24	24.13	**23.62**
	CS†	38.76	32.18	24.18	23.30
	LN+CS	39.59	32.73	24.24	23.32
	LN+CP+CS	**39.62**	**32.75**	**24.27**	23.30
b=100	base	35.17	28.48	23.54	23.50
	LN	38.60	31.97	24.04	23.14
	CP	37.64	30.82	23.77	23.65
	CP†	34.77	27.45	23.69	23.63
	LN+CP	38.93	32.39	23.95	23.60
	CS	39.60	32.71	24.01	**23.84**
	CS†	37.79	31.57	23.99	23.75
	LN+CS	39.88	33.20	24.22	23.60
	LN+CP+CS	**39.90**	**33.23**	**24.24**	23.65
b=500	base	23.40	17.95	23.15	23.24
	LN	37.60	30.81	23.95	23.16
	CP	34.81	28.82	23.43	23.46
	CP†	32.23	25.09	23.65	23.61
	LN+CP	37.88	31.46	23.77	23.64
	CS	39.50	32.77	23.96	**23.85**
	CS†	35.89	29.92	23.75	23.70
	LN+CS	**39.77**	**32.89**	**24.17**	23.57
	LN+CP+CS	39.73	32.85	24.17	23.69

Table 1: BLEU results of NMT systems. `base` = base system, `LN` = length normalization, `CP` = coverage penalty, and `CS` = our coverage score.

For comparison, we re-implemented the length normalization (LN) and coverage penalty (CP) methods (Wu et al., 2016). We used grid search to tune all hyperparameters on the development set as Wu et al. (2016). Specifically, weights for both CP and our CS are evaluated in interval $[0, 1]$ with step 0.1, while the weight for LN is in interval $[0.5, 1.5]$. We found that the settings determined with beam size 10 can be reliably applied to larger beam sizes in the preliminary experiments and thus we tuned all systems with beam size 10. For Chinese-English translation, we used a weight of 1.0 for both LN and CP, and set $\alpha = 0.6$ and $\beta = 0.4$. For English-German translation, we set the weights of LN and CP to 1.5 and 0.3, and set $\alpha = 0.3$ and $\beta = 0.2$. More details can be found in the Appendix.

3.2 Results

Table 1 shows the BLEU scores of the systems under different beam sizes (10, 100, and 500). We see, first of all, that our method outperforms

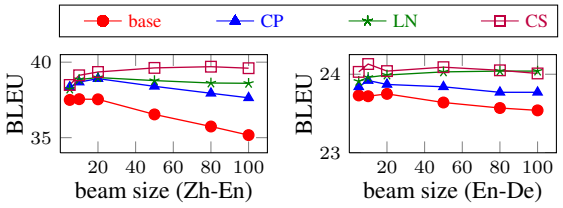

Figure 2: BLEU against beam size.

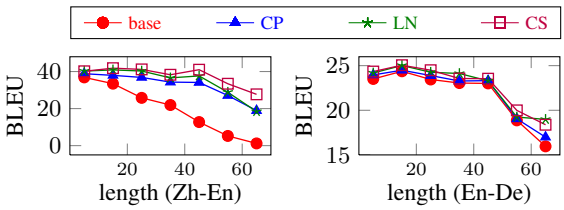

Figure 3: BLEU against sentence length.

four of the baselines, and the improvement is the largest when the beam size is 500. For a clear presentation, we plotted the BLEU curves by varying beam size. Figure 2 shows that our method has a consistent improvement as the beam size becomes larger, while others start to decline when the beam size is around 50, which indicates that integrating our coverage score into decoding is beneficial to prune out undesirable hypotheses when we search in a larger hypothesis space. We also see that the model gives even better results (+0.5 BLEU) after combining all these methods, which implies that our method doesn't overlap with the others. More interestingly, it is observed that the improvement on the En-De task is smaller than that on the Zh-En task. A possible reason is that there are relatively good word correspondences between English and German, and it is not so difficult for the base model to learn word deletions and insertions in En-De translation. Hence, the baseline system generates translations with proper lengths and does not benefit too much from the coverage model.

An interesting phenomenon in Table 1 is that using large beam size 100 rather than standard beam size (around 10) could give considerable improvements, e.g., 0.5 BLEU for Zh-En and 0.2 for En-De, yet the extremely large beam size 500 does not help much. This might result from the fact that our method is applied to each decoding step, thus helps model to search in a larger space and select better hypotheses, while a much larger beam size does not provide more benefits because the model already generates sufficiently good translations with a small beam size.

We also compared CP with our method by applying CP to each decoding step (Line CP†) and our method only to reranking (Line CS†) in Table 1. We noted that model performance dropped in most cases when CP was applied to each decoding step, and our method was helpful in reranking and obtained even better results as well when it is employed by beam search. This implies that the way of truncation is essential to enable the effective utilization of coverage inside beam search to achieve more significant improvements.

Entry		Zh-En			En-De		
		Len	Diff	LR	Len	Diff	LR
b=50	base	22.71	3.68	0.86	19.90	2.02	0.94
	CS	25.19	1.82	0.94	20.09	1.88	0.95
b=500	base	15.88	10.12	0.61	19.53	2.32	0.92
	CS	25.20	1.86	0.94	20.04	1.91	0.94

Table 2: Length statistics. `Len` = average length of translations, `Diff` = average length difference between translations and shortest references, `LR` = translation length ratio.

	$\beta = 0.0$	$\beta = 0.2$	$\beta = 0.4$	$\beta = 0.6$
$\alpha = 0.1$	36.2 / 23.7	37.8 / 24.0	37.8 / 24.0	37.7 / 23.9
$\alpha = 0.3$	30.8 / 18.9	38.2 / **24.1**	38.2 / 24.0	37.8 / 23.9
$\alpha = 0.6$	22.5 / 13.4	37.6 / 23.8	**39.1** / 23.9	38.6 / 23.8
$\alpha = 0.9$	13.0 / 7.03	26.6 / 17.2	35.1 / 21.6	35.4 / 21.7

Table 3: BLEU against α and β (zh-en/en-de)

Then, Figure 3 shows that our method has a relatively better ability to handle longer sentences. It obtains a significant improvement over the baselines when we translate sentences of more than 50 words. This is expectable because the coverage provides rich information from the past, which helps to address the long term dependency issue.

Another interesting question is whether the NMT systems can generate translations with appropriate lengths. To seek its answer, we studied the length difference between the MT output and the shortest reference. Table 2 shows that our method helps on both tasks. It generates translations whose lengths are closer to those of their references, which agrees with the BLEU results in Table 1. This is reasonable because our method encourages the hypotheses with higher coverage scores and thus higher recall. It means that our method can help the model to preserve the meaning of source words, which alleviates the under-translation problem.

Sensitivity analysis on α and β in Table 3 shows that the two tasks have different optimal choices of these values, which might be due to the natural need of length preference for different languages.

4 Related Work

The length preference and coverage problems have been discussed for years since the rise of statistical machine translation (Koehn, 2009). In NMT, several good methods have been developed. The simplest of these is length normalization which penalizes short translations in decoding (Wu et al., 2016). More sophisticated methods focus on modeling the coverage problem with extra sub-modules in NMT and require a training process (Tu et al., 2016; Mi et al., 2016).

Perhaps the most related work to this paper is Wu et al. (2016). In their work, the coverage problem can be interpreted in a probability story. However, it fails to account for the cases that one source word is translated into multiple target words and is thus of a total attention score > 1. To address this issue, we remove the probability constraint and make the coverage score interpretable for different cases. Another difference lies in that our coverage model is applied to every beam search step, while Wu et al. (2016)'s model affects only a small number of translation outputs.

Previous work have pointed out that BLEU scores of NMT systems drop as beam size increases (Britz et al., 2017; Tu et al., 2017; Koehn and Knowles, 2017), and the existing length normalization and coverage models can alleviate this problem to some extent. In this work we show that our method can do this much better. Almost no BLEU drop is observed even when beam size is set to 500.

5 Conclusion

We have described a coverage score and integrated it into a state-of-the-art NMT system. Our method is easy to implement and does not need training for additional models. Also, it performs well in searching with large beam sizes. On Chinese-English and English-German translation tasks, it outperforms several baselines significantly.

Acknowledgments

This work was supported in part by the National Science Foundation of China (61672138, 61432013 and 61671070), the Opening Project of Beijing Key Laboratory of Internet Culture and Digital Dissemination Research and the Fundamental Research Funds for the Central Universities. The authors would like to thank anonymous reviewers, Chunliang Zhang, Quan Du and Jingbo Zhu for their comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *International Conference on Learning Representations*.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc V. Le. 2017. Massive exploration of neural machine translation architectures. *CoRR*, abs/1703.03906.

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 the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *International Conference on Learning Representations*.

Philip Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 28–39.

Minh-Thang Luong, Eugene Brevdo, and Rui Zhao. 2017. Neural machine translation (seq2seq) tutorial. *https://github.com/tensorflow/nmt*.

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. Association for Computational Linguistics.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 955–960, Austin, Texas. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017. Neural machine translation with reconstruction. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA.*, pages 3097–3103.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 76–85, Berlin, Germany. 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, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Tong Xiao, Jingbo Zhu, Hao Zhang, and Qiang Li. 2012. Niutrans: An open source toolkit for phrase-based and syntax-based machine translation. In *Proceedings of the ACL 2012 System Demonstrations*, pages 19–24, Jeju Island, Korea. Association for Computational Linguistics.