Modeling Coverage for Neural Machine Translation

Zhaopeng Tu[†] **Zhengdong Lu**[†] **Yang Liu**[‡] **Xiaohua Liu**[†] **Hang Li**[†]

[†]Noah's Ark Lab, Huawei Technologies, Hong Kong
{tu.zhaopeng,lu.zhengdong,liuxiaohua3,hangli.hl}@huawei.com
[‡]Department of Computer Science and Technology, Tsinghua University, Beijing
liuyang2011@tsinghua.edu.cn

Abstract

Attention mechanism has enhanced state-of-the-art Neural Machine Translation (NMT) by jointly learning to align and translate. It tends to ignore past alignment information, however, which often leads to over-translation and under-translation. To address this problem, we propose coverage-based NMT in this paper. We maintain a coverage vector to keep track of the attention history. The coverage vector is fed to the attention model to help adjust future attention, which lets NMT system to consider more about untranslated source words. Experiments show that the proposed approach significantly improves both translation quality and alignment quality over standard attention-based NMT.[1]

1 Introduction

The past several years have witnessed the rapid progress of end-to-end Neural Machine Translation (NMT) (Sutskever et al., 2014; Bahdanau et al., 2015). Unlike conventional Statistical Machine Translation (SMT) (Koehn et al., 2003; Chiang, 2007), NMT uses a single and large neural network to model the entire translation process. It enjoys the following advantages. First, the use of distributed representations of words can alleviate the curse of dimensionality (Bengio et al., 2003). Second, there is no need to explicitly design features to capture translation regularities, which is quite difficult in SMT. Instead, NMT is capable of learning representations directly from the training data. Third, Long Short-Term Memory (Hochreiter and Schmidhuber, 1997) enables NMT to capture long-distance reordering, which is a significant challenge in SMT.

NMT has a serious problem, however, namely lack of *coverage*. In phrase-based SMT (Koehn et al., 2003), a decoder maintains a coverage vector to indicate whether a source word is translated or not. This is important for ensuring that each source word is translated in decoding. The decoding process is completed when all source words are "covered" or translated. In NMT, there is no such coverage vector and the decoding process ends only when the end-of-sentence mark is produced. We believe that lacking coverage might result in the following problems in conventional NMT:

1. Over-translation: some words are unnecessarily translated for multiple times;

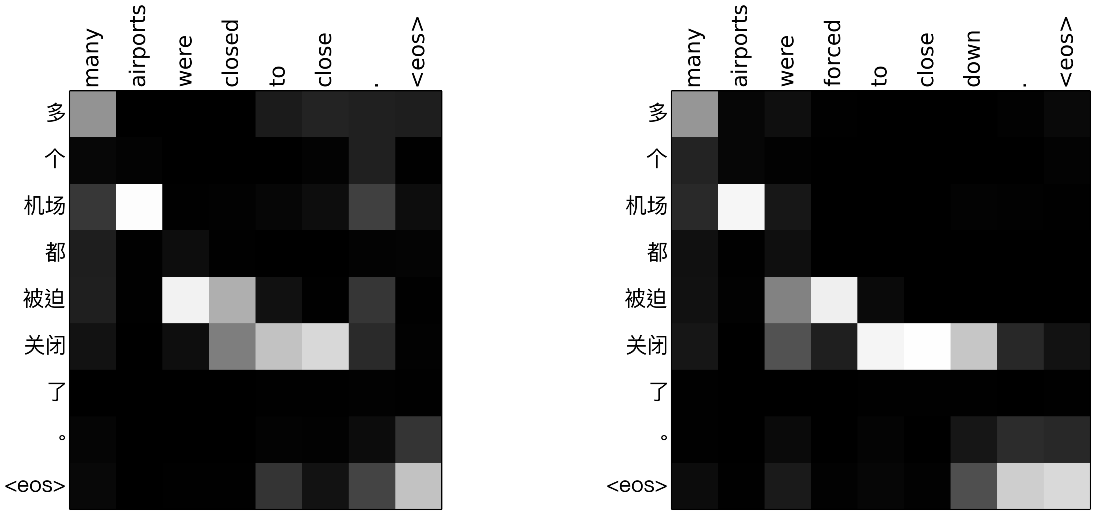
2. Under-translation: some words are mistakenly untranslated.

Specifically, in the state-of-the-art attention-based NMT model (Bahdanau et al., 2015), generating a target word heavily depends on the relevant parts of the source sentence, and a source word is involved in generation of all target words. As a result, over-translation and under-translation inevitably happen because of ignoring the "coverage" of source words (i.e., number of times a source word is translated to a target word). Figure 1(a) shows an example: the Chinese word "*guānbì*" is over translated to "*close*(*d*)" twice, while "*bèipò*" (means "*be forced to*") is mistakenly untranslated.

In this work, we propose a coverage mechanism to NMT (NMT-COVERAGE) to alleviate the over-translation and under-translation problems. Basically, we append a coverage vector to the intermediate representations of an NMT model, which are sequentially updated after each attentive read

[1] Our code is publicly available at https://github.com/tuzhaopeng/NMT-Coverage.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 76–85,
Berlin, Germany, August 7-12, 2016. ©2016 Association for Computational Linguistics



(a) Over-translation and under-translation generated by NMT.

(b) Coverage model alleviates the problems of over-translation and under-translation.

Figure 1: Example translations of (a) NMT without coverage, and (b) NMT with coverage. In conventional NMT without coverage, the Chinese word "*guānbì*" is over translated to "*close*(*d*)" twice, while "*bèipò*" (means "*be forced to*") is mistakenly untranslated. Coverage model alleviates these problems by tracking the "coverage" of source words.

during the decoding process, to keep track of the attention history. The coverage vector, when entering into attention model, can help adjust the future attention and significantly improve the overall alignment between the source and target sentences. This design contains many particular cases for coverage modeling with contrasting characteristics, which all share a clear linguistic intuition and yet can be trained in a data driven fashion. Notably, we achieve significant improvement even by simply using the sum of previous alignment probabilities as coverage for each word, as a successful example of incorporating linguistic knowledge into neural network based NLP models.

Experiments show that NMT-COVERAGE significantly outperforms conventional attention-based NMT on both translation and alignment tasks. Figure 1(b) shows an example, in which NMT-COVERAGE alleviates the over-translation and under-translation problems that NMT without coverage suffers from.

2 Background

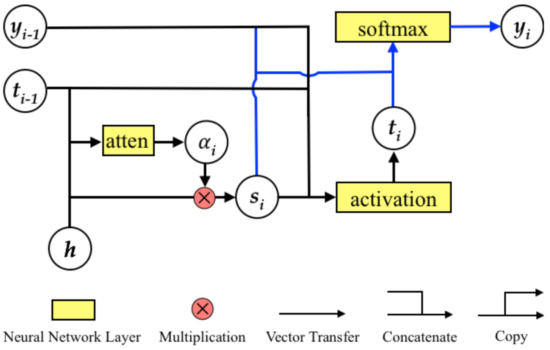


Figure 2: Architecture of attention-based NMT. Whenever possible, we omit the source index j to make the illustration less cluttered.

Our work is built on attention-based NMT (Bahdanau et al., 2015), which simultaneously conducts dynamic alignment and generation of the target sentence, as illustrated in Figure 2. It produces the translation by generating one target word y_i at each time step. Given an input sentence $\mathbf{x} = \{x_1, \ldots, x_J\}$ and previously generated words $\{y_1, \ldots, y_{i-1}\}$, the probability of generating next word y_i is

$$P(y_i|y_{<i}, \mathbf{x}) = softmax\big(g(y_{i-1}, \mathbf{t}_i, \mathbf{s}_i)\big) \quad (1)$$

where g is a non-linear function, and $\mathbf{t}_i$ is a decoding state for time step i, computed by

$$\mathbf{t}_i = f(\mathbf{t}_{i-1}, y_{i-1}, \mathbf{s}_i) \quad (2)$$

Here the activation function $f(\cdot)$ is a Gated Recurrent Unit (GRU) (Cho et al., 2014b), and $\mathbf{s}_i$ is a distinct source representation for time i, calculated as a weighted sum of the source annotations:

$$\mathbf{s}_i = \sum_{j=1}^{J} \alpha_{i,j} \cdot \mathbf{h}_j \tag{3}$$

where $\mathbf{h}_j = [\overrightarrow{h}_j^\top; \overleftarrow{h}_j^\top]^\top$ is the annotation of x_j from a bi-directional Recurrent Neural Network (RNN) (Schuster and Paliwal, 1997), and its weight $\alpha_{i,j}$ is computed by

$$\alpha_{i,j} = \frac{\exp(e_{i,j})}{\sum_{k=1}^{J} \exp(e_{i,k})} \tag{4}$$

and

$$\begin{aligned} e_{i,j} &= a(\mathbf{t}_{i-1}, \mathbf{h}_j) \\ &= v_a^\top \tanh(W_a \mathbf{t}_{i-1} + U_a \mathbf{h}_j) \end{aligned} \tag{5}$$

is an *attention model* that scores how well y_i and $\mathbf{h}_j$ match. With the attention model, it avoids the need to represent the entire source sentence with a single vector. Instead, the decoder selects parts of the source sentence to pay attention to, thus exploits an *expected annotation* $\mathbf{s}_i$ over possible alignments $\alpha_{i,j}$ for each time step i.

However, the attention model fails to take advantage of past alignment information, which is found useful to avoid over-translation and under-translation problems in conventional SMT (Koehn et al., 2003). For example, if a source word is translated in the past, it is less likely to be translated again and should be assigned a lower alignment probability.

3 Coverage Model for NMT

In SMT, a coverage set is maintained to keep track of which source words have been translated ("covered") in the past. Let us take $\mathbf{x} = \{x_1, x_2, x_3, x_4\}$ as an example of input sentence. The initial coverage set is $\mathcal{C} = \{0, 0, 0, 0\}$ which denotes that no source word is yet translated. When a translation rule $bp = (x_2 x_3, y_m y_{m+1})$ is applied, we produce one hypothesis labelled with coverage $\mathcal{C} = \{0, 1, 1, 0\}$. It means that the second and third source words are translated. The goal is to generate translation with full coverage $\mathcal{C} = \{1, 1, 1, 1\}$. A source word is translated when it is covered by one translation rule, and it is not allowed to be translated again in the future (i.e., *hard coverage*). In this way, each source word is guaranteed to be translated and only be translated once. As shown, coverage is essential for SMT since it avoids gaps and overlaps in translation of source words.

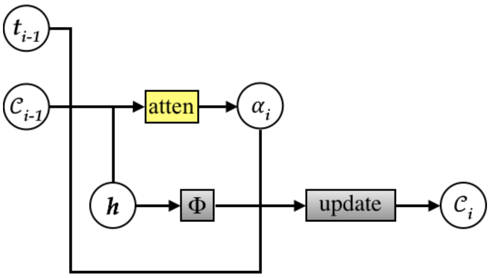


Figure 3: Architecture of coverage-based attention model. A coverage vector $\mathcal{C}_{i-1}$ is maintained to keep track of which source words have been translated before time i. Alignment decisions α_i are made jointly taking into account past alignment information embedded in $\mathcal{C}_{i-1}$, which lets the attention model to consider more about untranslated source words.

Modeling coverage is also important for attention-based NMT models, since they generally lack a mechanism to indicate whether a certain source word has been translated, and therefore are prone to the "coverage" mistakes: some parts of source sentence have been translated more than once or not translated. For NMT models, directly modeling coverage is less straightforward, but the problem can be significantly alleviated by keeping track of the attention signal during the decoding process. The most natural way for doing that would be to append a coverage vector to the annotation of each source word (i.e., $\mathbf{h}_j$), which is initialized as a zero vector but updated after every attentive read of the corresponding annotation. The coverage vector is fed to the attention model to help adjust future attention, which lets NMT system to consider more about untranslated source words, as illustrated in Figure 3.

3.1 Coverage Model

Since the coverage vector summarizes the attention record for $\mathbf{h}_j$ (and therefore for a small neighbor centering at the j^{th} source word), it will discourage further attention to it if it has been heavily attended, and implicitly push the attention to the less attended segments of the source sentence since the attention weights are normalized to one. This can potentially solve both coverage mistakes mentioned above, when modeled and learned properly.

Formally, the coverage model is given by

$$\mathcal{C}_{i,j} = g_{update}\big(\mathcal{C}_{i-1,j}, \alpha_{i,j}, \Phi(\mathbf{h}_j), \Psi\big) \quad (6)$$

where

- $g_{update}(\cdot)$ is the function that updates $\mathcal{C}_{i,j}$ after the new attention $\alpha_{i,j}$ at time step i in the decoding process;
- $\mathcal{C}_{i,j}$ is a d-dimensional coverage vector summarizing the history of attention till time step i on $\mathbf{h}_j$;
- $\Phi(\mathbf{h}_j)$ is a word-specific feature with its own parameters;
- Ψ are auxiliary inputs exploited in different sorts of coverage models.

Equation 6 gives a rather general model, which could take different function forms for $g_{update}(\cdot)$ and $\Phi(\cdot)$, and different auxiliary inputs Ψ (e.g., previous decoding state $\mathbf{t}_{i-1}$). In the rest of this section, we will give a number of representative implementations of the coverage model, which either leverage more linguistic information (Section 3.1.1) or resort to the flexibility of neural network approximation (Section 3.1.2).

3.1.1 Linguistic Coverage Model

We first consider at linguistically inspired model which has a small number of parameters, as well as clear interpretation. While the linguistically-inspired coverage in NMT is similar to that in SMT, there is one key difference: it indicates what percentage of source words have been translated (i.e., *soft coverage*). In NMT, each target word y_i is generated from all source words with probability $\alpha_{i,j}$ for source word x_j. In other words, the source word x_j is involved in generating all target words and the probability of generating target word y_i at time step i is $\alpha_{i,j}$. Note that unlike in SMT in which each source word is *fully translated* at one decoding step, the source word x_j is *partially translated* at each decoding step in NMT. Therefore, the coverage at time step i denotes the translated ratio of that each source word is translated.

We use a scalar ($d = 1$) to represent linguistic coverage for each source word and employ an accumulate operation for g_{update}. The initial value of linguistic coverage is zero, which denotes that the corresponding source word is not translated yet. We iteratively construct linguistic coverages through accumulation of alignment probabilities generated by the attention model, each of which is normalized by a distinct context-dependent weight. The coverage of source word x_j at time step i is computed by

$$\mathcal{C}_{i,j} = \mathcal{C}_{i-1,j} + \frac{1}{\Phi_j}\alpha_{i,j} = \frac{1}{\Phi_j}\sum_{k=1}^{i}\alpha_{k,j} \quad (7)$$

where Φ_j is a pre-defined weight which indicates the number of target words x_j is expected to generate. The simplest way is to follow Xu et al. (2015) in image-to-caption translation to fix $\Phi = 1$ for all source words, which means that we directly use the sum of previous alignment probabilities without normalization as coverage for each word, as done in (Cohn et al., 2016).

However, in machine translation, different types of source words may contribute differently to the generation of target sentence. Let us take the sentence pairs in Figure 1 as an example. The noun in the source sentence "*jīchǎng*" is translated into one target word "*airports*", while the adjective "*bèipò*" is translated into three words "*were forced to*". Therefore, we need to assign a distinct Φ_j for each source word. Ideally, we expect $\Phi_j = \sum_{i=1}^{I}\alpha_{i,j}$ with I being the total number of time steps in decoding. However, such desired value is not available before decoding, thus is not suitable in this scenario.

Fertility To predict Φ_j, we introduce the concept of *fertility*, which is firstly proposed in word-level SMT (Brown et al., 1993). Fertility of source word x_j tells how many target words x_j produces. In SMT, the fertility is a random variable Φ_j, whose distribution $p(\Phi_j = \phi)$ is determined by the parameters of word alignment models (e.g., IBM models). In this work, we simplify and adapt fertility from the original model and compute the fertility Φ_j by[2]

$$\Phi_j = \mathcal{N}(x_j|\mathbf{x}) = N \cdot \sigma(U_f \mathbf{h}_j) \quad (8)$$

where $N \in \mathbb{R}$ is a predefined constant to denote the maximum number of target words one source

[2] Fertility in SMT is a random variable with a set of fertility probabilities, $n(\Phi_j|x_j) = p(\Phi_{<j}, \mathbf{x})$, which depends on the fertilities of previous source words. To simplify the calculation and adapt it to the attention model in NMT, we define the fertility in NMT as a constant number, which is independent of previous fertilities.

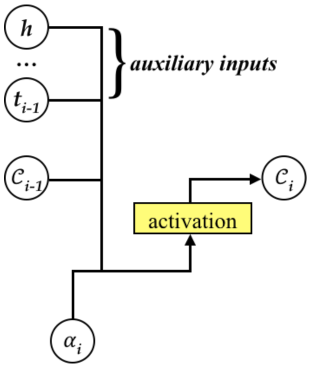


Figure 4: NN-based coverage model.

word can produce, $\sigma(\cdot)$ is a logistic sigmoid function, and $U_f \in \mathbb{R}^{1\times 2n}$ is the weight matrix. Here we use $\mathbf{h}_j$ to denote $(x_j|\mathbf{x})$ since $\mathbf{h}_j$ contains information about the whole input sentence with a strong focus on the parts surrounding x_j (Bahdanau et al., 2015). Since Φ_j does not depend on i, we can pre-compute it before decoding to minimize the computational cost.

3.1.2 Neural Network Based Coverage Model

We next consider Neural Network (NN) based coverage model. When $\mathcal{C}_{i,j}$ is a vector ($d > 1$) and $g_{update}(\cdot)$ is a neural network, we actually have an RNN model for coverage, as illustrated in Figure 4. In this work, we take the following form:

$$\mathcal{C}_{i,j} = f(\mathcal{C}_{i-1,j}, \alpha_{i,j}, \mathbf{h}_j, \mathbf{t}_{i-1})$$

where $f(\cdot)$ is a nonlinear activation function and $\mathbf{t}_{i-1}$ is the auxiliary input that encodes past translation information. Note that we leave out the word-specific feature function $\Phi(\cdot)$ and only take the input annotation $\mathbf{h}_j$ as the input to the coverage RNN. It is important to emphasize that the NN-based coverage model is able to be fed with arbitrary inputs, such as the previous attentional context $\mathbf{s}_{i-1}$. Here we only employ $\mathcal{C}_{i-1,j}$ for past alignment information, $\mathbf{t}_{i-1}$ for past translation information, and $\mathbf{h}_j$ for word-specific bias.[3]

Gating The neural function $f(\cdot)$ can be either a simple activation function tanh or a gating function that proves useful to capture long-distance dependencies. In this work, we adopt GRU for the gating activation since it is simple yet powerful (Chung et al., 2014). Please refer to (Cho et al., 2014b) for more details about GRU.

Discussion Intuitively, the two types of models summarize coverage information in "different languages". Linguistic models summarize coverage information in human language, which has a clear interpretation to humans. Neural models encode coverage information in "neural language", which can be "understood" by neural networks and let them to decide how to make use of the encoded coverage information.

3.2 Integrating Coverage into NMT

Although attention based model has the capability of jointly making alignment and translation, it does not take into consideration translation history. Specifically, a source word that has significantly contributed to the generation of target words in the past, should be assigned lower alignment probabilities, which may not be the case in attention based NMT. To address this problem, we propose to calculate the alignment probabilities by incorporating past alignment information embedded in the coverage model.

Intuitively, at each time step i in the decoding phase, coverage from time step $(i-1)$ serves as an additional input to the attention model, which provides complementary information of that how likely the source words are translated in the past. We expect the coverage information would guide the attention model to focus more on untranslated source words (i.e., assign higher alignment probabilities). In practice, we find that the coverage model does fulfill the expectation (see Section 5). The translated ratios of source words from linguistic coverages negatively correlate to the corresponding alignment probabilities.

More formally, we rewrite the attention model in Equation 5 as

$$\begin{aligned} e_{i,j} &= a(\mathbf{t}_{i-1}, \mathbf{h}_j, \mathcal{C}_{i-1,j}) \\ &= v_a^\top \tanh(W_a \mathbf{t}_{i-1} + U_a \mathbf{h}_j + V_a \mathcal{C}_{i-1,j}) \end{aligned}$$

where $\mathcal{C}_{i-1,j}$ is the coverage of source word x_j before time i. $V_a \in \mathbb{R}^{n\times d}$ is the weight matrix for coverage with n and d being the numbers of hidden units and coverage units, respectively.

[3] In our preliminary experiments, considering more inputs (e.g., current and previous attentional contexts, unnormalized attention weights $e_{i,j}$) does not always lead to better translation quality. Possible reasons include: 1) the inputs contains duplicate information, and 2) more inputs introduce more back-propagation paths and therefore make it difficult to train. In our experience, one principle is to only feed the coverage model inputs that contain distinct information, which are complementary to each other.

4 Training

We take end-to-end learning for the NMT-COVERAGE model, which learns not only the parameters for the "original" NMT (i.e., θ for encoding RNN, decoding RNN, and attention model) but also the parameters for coverage modeling (i.e., η for annotation and guidance of attention) . More specifically, we choose to maximize the likelihood of reference sentences as most other NMT models (see, however (Shen et al., 2016)):

$$(\theta^*, \eta^*) = \arg\max_{\theta,\eta} \sum_{n=1}^{N} \log P(\mathbf{y}_n|\mathbf{x}_n; \theta, \eta) \quad (9)$$

No auxiliary objective For the coverage model with a clearer linguistic interpretation (Section 3.1.1), it is possible to inject an auxiliary objective function on some intermediate representation. More specifically, we may have the following objective:

$$(\theta^*, \eta^*) = \arg\max_{\theta,\eta} \sum_{n=1}^{N} \Big\{ \log P(\mathbf{y}_n|\mathbf{x}_n; \theta, \eta) - \lambda \Big\{ \sum_{j=1}^{J} (\Phi_j - \sum_{i=1}^{I} \alpha_{i,j})^2; \eta \Big\} \Big\}$$

where the term $\{ \sum_{j=1}^{J} (\Phi_j - \sum_{i=1}^{I} \alpha_{i,j})^2; \eta \}$ penalizes the discrepancy between the sum of alignment probabilities and the expected fertility for linguistic coverage. This is similar to the more explicit training for fertility as in Xu et al. (2015), which encourages the model to pay equal attention to every part of the image (i.e., $\Phi_j = 1$). However, our empirical study shows that the combined objective consistently worsens the translation quality while slightly improves the alignment quality.

Our training strategy poses less constraints on the dependency between Φ_j and the attention than a more explicit strategy taken in (Xu et al., 2015). We let the objective associated with the translation quality (i.e., the likelihood) to drive the training, as in Equation 9. This strategy is arguably advantageous, since the attention weight on a hidden state $\mathbf{h}_j$ cannot be interpreted as the proportion of the corresponding word being translated in the target sentence. For one thing, the hidden state $\mathbf{h}_j$, after the transformation from encoding RNN, bears the contextual information from other parts of the source sentence, and thus loses the rigid correspondence with the corresponding word. Therefore, penalizing the discrepancy between the sum of alignment probabilities and the expected fertility does not hold in this scenario.

5 Experiments

5.1 Setup

We carry out experiments on a Chinese-English translation task. Our training data for the translation task consists of 1.25M sentence pairs extracted from LDC corpora[4] , with 27.9M Chinese words and 34.5M English words respectively. We choose NIST 2002 dataset as our development set, and the NIST 2005, 2006 and 2008 datasets as our test sets. We carry out experiments of the alignment task on the evaluation dataset from (Liu and Sun, 2015), which contains 900 manually aligned Chinese-English sentence pairs. We use the case-insensitive 4-gram NIST BLEU score (Papineni et al., 2002) for the translation task, and the alignment error rate (AER) (Och and Ney, 2003) for the alignment task. To better estimate the quality of the soft alignment probabilities generated by NMT, we propose a variant of AER, naming *SAER*:

$$SAER = 1 - \frac{|M_A \times M_S| + |M_A \times M_P|}{|M_A| + |M_S|}$$

where A is a candidate alignment, and S and P are the sets of sure and possible links in the reference alignment respectively ($S \subseteq P$). M denotes alignment matrix, and for both M_S and M_P we assign the elements that correspond to the existing links in S and P with probabilities 1 while assign the other elements with probabilities 0. In this way, we are able to better evaluate the quality of the soft alignments produced by attention-based NMT. We use *sign-test* (Collins et al., 2005) for statistical significance test.

For efficient training of the neural networks, we limit the source and target vocabularies to the most frequent 30K words in Chinese and English, covering approximately 97.7% and 99.3% of the two corpora respectively. All the out-of-vocabulary words are mapped to a special token UNK. We set $N = 2$ for the fertility model in the linguistic coverages. We train each model with the sentences of length up to 80 words in the training data. The word embedding dimension is 620 and the size of a hidden layer is 1000. All the other settings are the same as in (Bahdanau et al., 2015).

[4] The corpora include LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

#	System	#Params	MT05	MT06	MT08	Avg.
1	Moses	–	31.37	30.85	23.01	28.41
2	GroundHog	84.3M	30.61	31.12	23.23	28.32
3	+ Linguistic coverage w/o fertility	+1K	$31.26^{\dagger}$	$32.16^{\dagger\ddagger}$	$24.84^{\dagger\ddagger}$	29.42
4	+ Linguistic coverage w/ fertility	+3K	$32.36^{\dagger\ddagger}$	$32.31^{\dagger\ddagger}$	$24.91^{\dagger\ddagger}$	29.86
5	+ NN-based coverage w/o gating ($d = 1$)	+4K	$31.94^{\dagger\ddagger}$	$32.11^{\dagger\ddagger}$	23.31	29.12
6	+ NN-based coverage w/ gating ($d = 1$)	+10K	$31.94^{\dagger\ddagger}$	$32.16^{\dagger\ddagger}$	$24.67^{\dagger\ddagger}$	29.59
7	+ NN-based coverage w/ gating ($d = 10$)	+100K	$\mathbf{32.73}^{\dagger\ddagger}$	$\mathbf{32.47}^{\dagger\ddagger}$	$\mathbf{25.23}^{\dagger\ddagger}$	**30.14**

Table 1: Evaluation of translation quality. d denotes the dimension of NN-based coverages, and † and ‡ indicate statistically significant difference ($p < 0.01$) from GroundHog and Moses, respectively. "+" is on top of the baseline system GroundHog.

We compare our method with two state-of-the-art models of SMT and NMT[5]:

- **Moses** (Koehn et al., 2007): an open source phrase-based translation system with default configuration and a 4-gram language model trained on the target portion of training data.
- **GroundHog** (Bahdanau et al., 2015): an attention-based NMT system.

5.2 Translation Quality

Table 1 shows the translation performances measured in BLEU score. Clearly the proposed NMT-COVERAGE significantly improves the translation quality in all cases, although there are still considerable differences among different variants.

Parameters Coverage model introduces few parameters. The baseline model (i.e., GroundHog) has 84.3M parameters. The linguistic coverage using fertility introduces 3K parameters (2K for fertility model), and the NN-based coverage with gating introduces 10K$\times d$ parameters (6K$\times d$ for gating), where d is the dimension of the coverage vector. In this work, the most complex coverage model only introduces 0.1M additional parameters, which is quite small compared to the number of parameters in the existing model (i.e., 84.3M).

Speed Introducing the coverage model slows down the training speed, but not significantly. When running on a single GPU device Tesla K80, the speed of the baseline model is 960 target words per second. System 4 ("+Linguistic coverage with fertility") has a speed of 870 words per second, while System 7 ("+NN-based coverage (d=10)") achieves a speed of 800 words per second.

Linguistic Coverages (Rows 3 and 4): Two observations can be made. First, the simplest linguistic coverage (Row 3) already significantly improves translation performance by 1.1 BLEU points, indicating that coverage information is very important to the attention model. Second, incorporating fertility model boosts the performance by better estimating the covered ratios of source words.

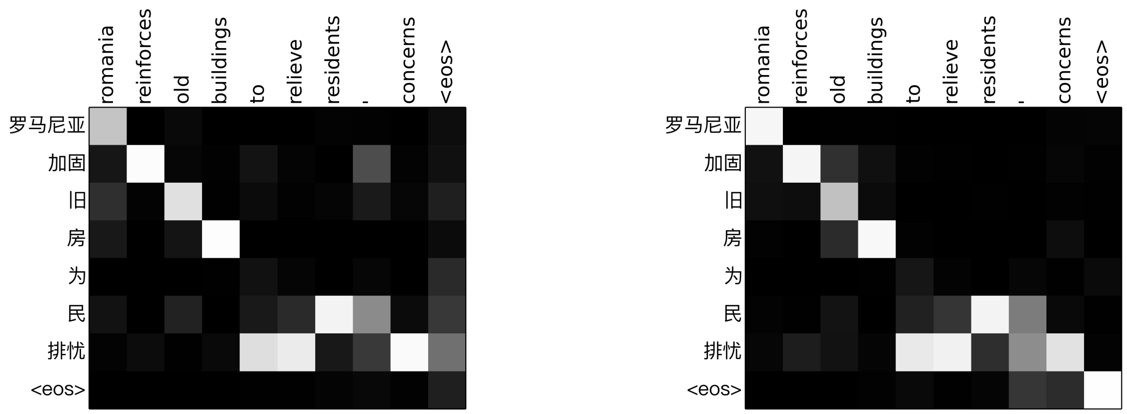
NN-based Coverages (Rows 5-7): (1) *Gating* (Rows 5 and 6): Both variants of NN-based coverages outperform GroundHog with averaged gains of 0.8 and 1.3 BLEU points, respectively. Introducing gating activation function improves the performance of coverage models, which is consistent with the results in other tasks (Chung et al., 2014). (2) *Coverage dimensions* (Rows 6 and 7): Increasing the dimension of coverage models further improves the translation performance by 0.6 point in BLEU score, at the cost of introducing more parameters (e.g., from 10K to 100K).[6]

5.3 Alignment Quality

Table 2 lists the alignment performances. We find that coverage information improves attention model as expected by maintaining an annotation summarizing attention history on each source word. More specifically, linguistic coverage with fertility significantly reduces alignment errors under both metrics, in which fertility plays an important role. NN-based coverages, however, does not significantly reduce alignment errors until increasing the coverage dimension from 1 to 10. It indicates that NN-based models need slightly more

[5] There are recent progress on aggregating multiple models or enlarging the vocabulary(e.g., in (Jean et al., 2015)), but here we focus on the generic models.

[6] In a pilot study, further increasing the coverage dimension only slightly improved the translation performance. One possible reason is that encoding the relatively simple coverage information does not require too many dimensions.



(a) Groundhog

(b) + NN cov. w/ gating $(d = 10)$

Figure 5: Example alignments. Using coverage mechanism, translated source words are less likely to contribute to generation of the target words next (e.g., top-right corner for the first four Chinese words.).

System	SAER	AER
GroundHog	67.00	54.67
+ Ling. cov. w/o fertility	66.75	53.55
+ Ling. cov. w/ fertility	64.85	52.13
+ NN cov. w/o gating $(d = 1)$	67.10	54.46
+ NN cov. w/ gating $(d = 1)$	66.30	53.51
+ NN cov. w/ gating $(d = 10)$	**64.25**	**50.50**

Table 2: Evaluation of alignment quality. The lower the score, the better the alignment quality.

dimensions to encode the coverage information.

Figure 5 shows an example. The coverage mechanism does meet the expectation: the alignments are more concentrated and most importantly, translated source words are less likely to get involved in generation of the target words next. For example, the first four Chinese words are assigned lower alignment probabilities (i.e., darker color) after the corresponding translation "*romania reinforces old buildings*" is produced.

5.4 Effects on Long Sentences

Following Bahdanau et al. (2015), we group sentences of similar lengths together and compute BLEU score and averaged length of translation for each group, as shown in Figure 6. Cho et al. (2014a) show that the performance of Groundhog drops rapidly when the length of input sentence increases. Our results confirm these findings. One main reason is that Groundhog produces much shorter translations on longer sentences (e.g., $>$ 40, see right panel in Figure 6), and thus faces a serious under-translation problem. NMT-COVERAGE alleviates this problem by incorporating coverage information into the attention model, which in general pushes the attention to untranslated parts of the source sentence and implicitly discourages early stop of decoding. It is worthy to emphasize that both NN-based coverages (with gating, $d = 10$) and linguistic coverages (with fertility) achieve similar performances on long sentences, reconfirming our claim that the two variants improve the attention model in their own ways.

As an example, consider this source sentence in the test set:

> *qiáodān běn sàijì píngjūn défēn 24.3fēn , tā zài sān zhōu qián jiēshòu shǒushù , qiúduì zài cǐ qījiān 4 shèng 8 fù .*

Groundhog translates this sentence into:

> *jordan achieved an average score of eight weeks ahead with a surgical operation three weeks ago .*

in which the sub-sentence "*, qiúduì zài cǐ qījiān 4 shèng 8 fù*" is under-translated. With the (NN-based) coverage mechanism, NMT-COVERAGE translates it into:

> *jordan 's average score points to UNK this year . he received surgery before three weeks , with a team in the period of 4 to 8 .*

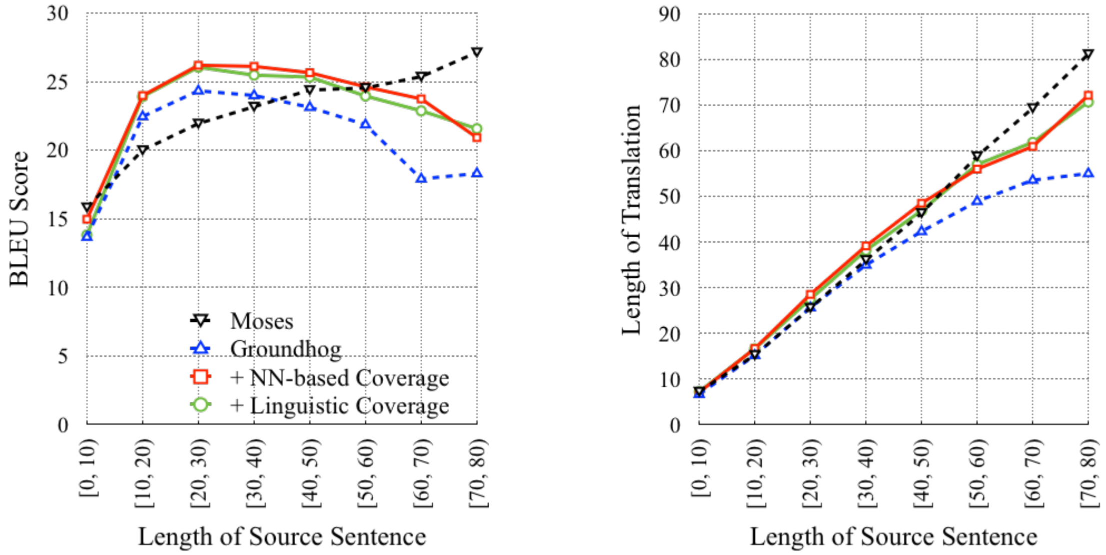


Figure 6: Performance of the generated translations with respect to the lengths of the input sentences. Coverage models alleviate under-translation by producing longer translations on long sentences.

in which the under-translation is rectified.

The quantitative and qualitative results show that the coverage models indeed help to alleviate under-translation, especially for long sentences consisting of several sub-sentences.

6 Related Work

Our work is inspired by recent works on improving attention-based NMT with techniques that have been successfully applied to SMT. Following the success of Minimum Risk Training (MRT) in SMT (Och, 2003), Shen et al. (2016) proposed MRT for end-to-end NMT to optimize model parameters directly with respect to evaluation metrics. Based on the observation that attention-based NMT only captures partial aspects of attentional regularities, Cheng et al. (2016) proposed agreement-based learning (Liang et al., 2006) to encourage bidirectional attention models to agree on parameterized alignment matrices. Along the same direction, inspired by the coverage mechanism in SMT, we propose a coverage-based approach to NMT to alleviate the over-translation and under-translation problems.

Independent from our work, Cohn et al. (2016) and Feng et al. (2016) made use of the concept of "fertility" for the attention model, which is similar in spirit to our method for building the linguistically inspired coverage with fertility. Cohn et al. (2016) introduced a feature-based fertility that includes the total alignment scores for the surrounding source words. In contrast, we make prediction of fertility before decoding, which works as a normalizer to better estimate the coverage ratio of each source word. Feng et al. (2016) used the previous attentional context to represent *implicit fertility* and passed it to the attention model, which is in essence similar to the input-feed method proposed in (Luong et al., 2015). Comparatively, we predict *explicit fertility* for each source word based on its encoding annotation, and incorporate it into the linguistic-inspired coverage for attention model.

7 Conclusion

We have presented an approach for enhancing NMT, which maintains and utilizes a coverage vector to indicate whether each source word is translated or not. By encouraging NMT to pay less attention to translated words and more attention to untranslated words, our approach alleviates the serious over-translation and under-translation problems that traditional attention-based NMT suffers from. We propose two variants of coverage models: *linguistic coverage* that leverages more linguistic information and *NN-based coverage* that resorts to the flexibility of neural network approximation . Experimental results show that both variants achieve significant improvements in terms of translation quality and alignment quality over NMT without coverage.

Acknowledgement

This work is supported by China National 973 project 2014CB340301. Yang Liu is supported by the National Natural Science Foundation of China (No. 61522204) and the 863 Program (2015AA011808). We thank the anonymous reviewers for their insightful comments.

References

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

[Bengio et al.2003] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. 2003. A neural probabilistic language model. *JMLR*.

[Brown et al.1993] Peter E. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

[Cheng et al.2016] Yong Cheng, Shiqi Shen, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Agreement-based Joint Training for Bidirectional Attention-based Neural Machine Translation. In *IJCAI 2016*.

[Chiang2007] David Chiang. 2007. Hierarchical phrase-based translation. *CL*.

[Cho et al.2014a] Kyunghyun Cho, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: encoder–decoder approaches. In *SSST 2014*.

[Cho et al.2014b] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP 2014*.

[Chung et al.2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv*.

[Cohn et al.2016] Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vylomova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating Structural Alignment Biases into an Attentional Neural Translation Model. In *NAACL 2016*.

[Collins et al.2005] Michael Collins, Philipp Koehn, and Ivona Kučerová. 2005. Clause restructuring for statistical machine translation. In *ACL 2005*.

[Feng et al.2016] Shi Feng, Shujie Liu, Mu Li, and Ming Zhou. 2016. Implicit distortion and fertility models for attention-based encoder-decoder nmt model. *arXiv*.

[Hochreiter and Schmidhuber1997] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*.

[Jean et al.2015] Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *ACL 2015*.

[Koehn et al.2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL 2003*.

[Koehn et al.2007] 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 *ACL 2007*.

[Liang et al.2006] Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *NAACL 2006*.

[Liu and Sun2015] Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *AAAI 2015*.

[Luong et al.2015] Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*.

[Och and Ney2003] Franz J. Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

[Och2003] Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL 2003*.

[Papineni et al.2002] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*.

[Schuster and Paliwal1997] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681.

[Shen et al.2016] Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum Risk Training for Neural Machine Translation. In *ACL 2016*.

[Sutskever et al.2014] Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*.

[Xu et al.2015] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In *ICML 2015*.