COVERAGE-BASED NEURAL MACHINE TRANSLATION

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

[†]Huawei Noah's Ark Lab, 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 advanced state-of-the-art neural machine translation (NMT) by jointly learning to align and translate. However, attentional NMT ignores past alignment information, which leads to over-translation and under-translation problems. In response to this problem, we maintain a coverage vector to keep track of the attention history. The coverage vector is fed to the attention model to help adjust the future attention, which guides NMT to pay more attention to the untranslated source words. Experiments show that coverage-based NMT significantly improves both translation and alignment qualities over NMT without coverage.

1 INTRODUCTION

The past several years have witnessed the rapid development of end-to-end neural machine translation (NMT) (Kalchbrenner & Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2015). Unlike conventional statistical machine translation (SMT) (Brown et al., 1993; Koehn et al., 2003), NMT proposes to use a single, large neural network instead of latent structures to model the translation process. However, a serious problem with NMT is the lack of *coverage*. In SMT, 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 exactly in decoding. The decoding process is completed when all source words are translated. In NMT, there is no such coverage vector and the decoding process ends only when the end-of-sentence tag is produced. We believe that lacking coverage might result in following problems in NMT:

1. Over-translation: some words are unnecessarily translated for multiple times;
2. Under-translation: some words are wrongly untranslated.

In this work, we propose a coverage-based approach to NMT to alleviate the over-translation and under-translation problems. Basically, we append annotation vectors to the intermediate representation of NMT models, which is updated after each attentive read during the decoding process to keep track of the attention history. Those annotation vectors, when entering into attention model, can help adjust the future attention and significantly improve the alignment between source and target. This design potentially 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, in a simple and effective case, we achieve by far the best performance by redefining the concept of fertility, as a successful example of re-introducing linguistic knowledge into neural network-based NLP models. Experiments on large-scale Chinese-English datasets show that our coverage-based NMT system outperforms conventional attentional NMT significantly on both translation and alignment tasks.

2 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. Take an input sentence $\mathbf{x} = \{x_1, x_2, x_3, x_4\}$ as an example, the initial coverage set is $\mathcal{C} = \{0, 0, 0, 0\}$ which denotes no source word is yet translated. When a translation

rule is used to translate $\{x_2, x_3\}$, 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. 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 overlap when translating source words.

For NMT, 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 is to append an annotation vector β_j to each h_j (the input annotation of the j^{th} source word), which is uniformly initialized but updated after every attentive read of the corresponding hidden state. This annotation vector will enter the soft attention model for alignment. Intuitively, at each time step i in the decoding phase, coverage from time step $(i-1)$ serves as input to the attention model, which provides complementary information of that how likely the source words are translated in the past. Since $\beta_{i-1,j}$ summarizes the attention record for h_j, 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 could potentially solve both coverage mistakes mentioned above, when modeled and learned properly.

Formally, the coverage model is given by

$$\beta_{i,j} = g_{update}\big(\beta_{i-1,j}, \alpha_{i,j}, \Phi(h_j), \Psi\big) \tag{1}$$

where $g_{update}(\cdot)$ is the function that updates $\beta_{i,j}$ after the new attention at time step i, $\beta_{i,j}$ is a d-dimensional annotation vector summarizing the history of attention till time step i on h_j, $\Phi_i(h_j)$ is a word-specific feature with its own parameters, and Ψ are auxiliary inputs exploited in different sorts of coverage models.

Equation 1 gives a rather general model, which could take different function forms for $g_{update}(\cdot)$ and $\Phi(\cdot)$, and different auxiliary inputs $auxs$ (e.g. previous decoding state s_{i-1}). In the rest of this section, we will give a number of representative implementations of the annotation model, which either resort to the flexibility of neural network function approximation (Section 2.1) or bear more linguistic intuition (Section 2.2).

2.1 Neural Network-based Coverage Model

When β_j is a vector ($d > 1$) and $g_{update}(\cdot)$ takes a neural network (NN) form, we actually have a recurrent neural network (RNN) model for annotation.In our work, we take the following form

$$\beta_{i,j} = f(\beta_{i-1,j}, \alpha_{i,j}, h_j, s_{i-1})$$

where the activation function $f(\cdot)$ is a gated recurrent unit (GRU) (Cho et al., 2014) and s_{i-1} is the auxiliary input that encodes past translation information. Note that we leave out the the word-specific feature function $\Phi(\cdot)$ and only take h_j as the input to the annotation RNN. It is important to emphasize that the NN-based annotation is able to be fed with arbitrary inputs, such as the previous attentional context c_{i-1}. Here we only employ α_{i-1} for past alignment information, s_{i-1} for past translation information, and h_j for word-specific bias.

Although the NN-based model enjoys the flexibility brought by the nonlinear form, its lack of clear linguistic meaning may render it hard to train: the annotation model can only be trained along with the attention model and get the supervision signal from it in back-propagation, which could be weak (relatively distant from the decoding process) and noisy (after the distortion from other under-trained components in the decoder RNN). Partially to overcome this problem, we also propose the linguistically inspired model which has much clearer interpretation but much less parameters.

2.2 Linguistic Coverage Model

While linguistically-inspired coverage in NMT is similar in spirit 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 probabilities $\alpha_{i,j}$ for source word x_j. In other words, each source word x_j involves in generating all target words and generates $\alpha_{i,j}$ target word at time step i. Note that unlike in SMT where each source word is not *fully translated* at

Table 1: Evaluation of translation and alignment qualities. Higher score means better translation quality, while lower score means better alignment quality. Linguistic coverage overall outperforms its NN-based counterpart on both translation and alignment tasks, indicating that explicitly linguistic regularities are very important to the attention model.

System	Translation	Alignment
Moses	28.41	–
NMT (Bahdanau et al., 2015)	26.20	56.78
NMT + NN-based coverage	27.14	56.17
NMT + Linguistic coverage	27.70	54.91

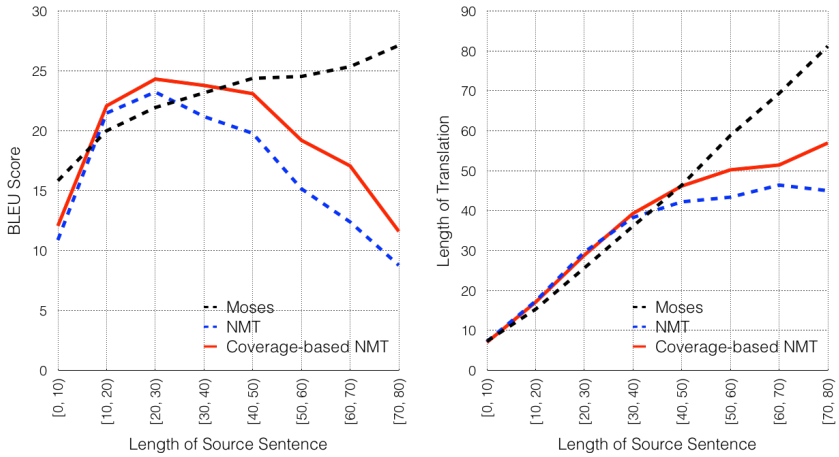

Figure 1: Performance of the generated translations on the test set with respect to the lengths of the input sentences. Coverage-based NMT alleviates the problem of under-translation on long sentences by producing longer translations, leading to better translation performances.

one decoding step, 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 coverages for each source word and employ an accumulate operation for g_{update}. We iteratively construct linguistic coverages through an 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

$$\beta_{i,j} = \frac{1}{\Phi_j} \sum_{k=1}^{i} \alpha_{k,j} \tag{2}$$

where Φ_j is a pre-defined weight which indicates the number of target words x_j is expected to generate. 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 this work, we simplify and adapt fertility from the original model[1] and compute the fertility Φ_j by

$$\Phi_j = \mathcal{N}(x_j|\mathbf{x}) = \mathcal{N}(h_j) = N \cdot \sigma(U_f h_j) \tag{3}$$

where $N \in \mathbb{R}$ is a predefined constantto denoting the maximum number of target words one source 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 h_j to denote $(x_j|\mathbf{x})$ since 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.

[1]Fertility in SMT is a random variable with a set of fertility probabilities, $n(\Phi_j|x_j) = p(\Phi_1^{j-1}, \mathbf{x})$, 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.

REFERENCES

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

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

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

Nal Kalchbrenner and Phil Blunsom. Recurrent continuous translation models. In *EMNLP 2013*, 2013.

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

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