Neural Machine Translation with Word Predictions

Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xinyu Dai and **Jiajun Chen**
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing 210023, China
{wengrx, huangsj, zhengzx, daixy, chenjj}@nlp.nju.edu.cn

Abstract

In the encoder-decoder architecture for neural machine translation (NMT), the hidden states of the recurrent structures in the encoder and decoder carry the crucial information about the sentence.These vectors are generated by parameters which are updated by back-propagation of translation errors through time. We argue that propagating errors through the end-to-end recurrent structures are not a direct way of control the hidden vectors. In this paper, we propose to use word predictions as a mechanism for direct supervision. More specifically, we require these vectors to be able to predict the vocabulary in target sentence. Our simple mechanism ensures better representations in the encoder and decoder without using any extra data or annotation. It is also helpful in reducing the target side vocabulary and improving the decoding efficiency. Experiments on Chinese-English and German-English machine translation tasks show BLEU improvements by 4.53 and 1.3, respectively.

1 Introduction

The encoder-decoder based neural machine translation (NMT) models (Sutskever et al., 2014; Cho et al., 2014) have been developing rapidly. Sutskever et al. (2014) propose to encode the source sentence as a fixed-length vector representation, based on which the decoder generates the target sequence, where both the encoder and decoder are recurrent neural networks (RNN) (Sutskever et al., 2014) or their variants (Cho et al., 2014; Chung et al., 2014; Bahdanau et al., 2014). In this framework, the fixed-length vector plays the crucial role of transitioning the information of the sentence from the source side to the target side.

Later, attention mechanisms are proposed to enhance the source side representations (Bahdanau et al., 2014; Luong et al., 2015b). The source side context is computed at each time-step of decoding, based on the attention weights between the source side representations and the current hidden state of the decoder. However, the hidden states in the recurrent decoder still originate from the single fixed-length representation (Luong et al., 2015b), or the average of the bi-directional representations (Bahdanau et al., 2014). Here we refer to the representation as *initial state*.

Interestingly, Britz et al. (2017) find that the value of initial state does not affect the translation performance, and prefer to set the initial state to be a zero vector. On the contrary, we argue that initial state still plays an important role of translation, which is currently neglected. We notice that beside the end-to-end error back propagation for the initial and transition parameters, there is no direct control of the initial state in the current NMT architectures. Due to the large number of parameters, it may be difficult for the NMT system to learn the proper sentence representation as the initial state. Thus, the model is very likely to get stuck in local minimums, making the translation process arbitrary and unstable.

In this paper, we propose to augment the current NMT architecture with a word prediction mechanism. More specifically, we require the initial state of the decoder to be able to predict all the words in the target sentence. In this way, there is a specific objective for learning the initial state. Thus the learnt source side representation will be better constrained. We further extend this idea by applying the word predictions mechanism to all the hidden states of the decoder. So the transition between different decoder states could be controlled

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 136–145
Copenhagen, Denmark, September 7–11, 2017.

as well.

Our mechanism is simple and requires no additional data or annotation. The proposed word predictions mechanism could be used as a training method and brings no extra computing cost during decoding.

Experiments on the Chinese-English and German-English translation tasks show that both the constraining of the initial state and the decoder hidden states bring significant improvement over the baseline systems. Furthermore, using the word prediction mechanism on the initial state as a word predictor to reduce the target side vocabulary could greatly improve the decoding efficiency, without a significant loss on the translation quality.

2 Related Work

Many previous works have noticed the problem of training an NMT system with lots of parameters. Some of them prefer to use the dropout technique (Srivastava et al., 2014; Luong et al., 2015b; Meng et al., 2016). Another possible choice is to ensemble several models with random starting points (Sutskever et al., 2014; Jean et al., 2015; Luong and Manning, 2016). Both techniques could bring more stable and better results. But they are general training techniques of neural networks, which are not specifically targeting the modeling of the translation process like ours. We will make empirical comparison with them in the experiments.

The way we add the word prediction is similar to the research of multi-task learning. Dong et al. (2015) propose to share an encoder between different translation tasks. Luong et al. (2015a) propose to jointly learn the translation task for different languages, the parsing task and the image captioning task, with a shared encoder or decoder. Zhang and Zong (2016) propose to use multitask learning for incorporating source side monolingual data. Different from these attempts, our method focuses solely on the current translation task, and does not require any extra data or annotation.

In the other sequence to sequence tasks, Suzuki and Nagata (2017) propose the idea for predicting words by using encoder information. However, the purpose and the way of our mechanism are different from them.

The word prediction technique has been applied in the research of both statistical machine translation (SMT) (Bangalore et al., 2007; Mauser et al., 2009; Jeong et al., 2010; Tran et al., 2014) and NMT (Mi et al., 2016; L'Hostis et al., 2016). In these research, word prediction mechanisms are employed to decide the selection of words or constrain the target vocabulary, while in this paper, we use word prediction as a control mechanism for neural model training.

3 Notations and Backgrounds

We present a popular NMT framework with the encoder-decoder architecture (Cho et al., 2014; Bahdanau et al., 2014) and the attention networks (Luong et al., 2015b), based on which we propose our word prediction mechanism.

Denote a source-target sentence pair as $\{\mathbf{x}, \mathbf{y}\}$ from the training set, where $\mathbf{x}$ is the source word sequence $(x_1, x_2, \cdots, x_{|\mathbf{x}|})$ and $\mathbf{y}$ is the target word sequence $(y_1, y_2, \cdots, y_{|\mathbf{y}|})$, $|\mathbf{x}|$ and $|\mathbf{y}|$ are the length of $\mathbf{x}$ and $\mathbf{y}$, respectively.

In the encoding stage, a bi-directional recurrent neural network is used (Bahdanau et al., 2014) to encode $\mathbf{x}$ into a sequence of vectors $(\mathbf{h}_1, \mathbf{h}_2, \cdots, \mathbf{h}_{|\mathbf{x}|})$. For each x_i, the representation $\mathbf{h}_i$ is:

$$\mathbf{h}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \tag{1}$$

where $[\cdot;\cdot]$ denotes the concatenation of column vectors; $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ denote the hidden vectors for the word x_i in the forward and backward RNNs, respectively.

The gated recurrent unit (GRU) is used as the recurrent unit in each RNN, which is shown to have promising results in speech recognition and machine translation (Cho et al., 2014). Formally, the hidden state $\mathbf{h}_i$ at time step i of the forward RNN encoder is defined by the GRU function $g_{\overrightarrow{e}}(\cdot,\cdot)$, as follows:

$$\begin{aligned} \overrightarrow{\mathbf{h}}_i &= g_{\overrightarrow{e}}(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{emb}_{x_i}) && (2)\\ &= (\mathbf{1} - \overrightarrow{\mathbf{z}}_i) \odot \overrightarrow{\mathbf{h}}_{i-1} + \overrightarrow{\mathbf{z}}_i \odot \overrightarrow{\mathbf{h}'}_i \\ \overrightarrow{\mathbf{z}}_i &= \sigma(\overrightarrow{\mathbf{W}}_z[\mathbf{emb}_{x_i}; \overrightarrow{\mathbf{h}}_{i-1}]) && (3)\\ \overrightarrow{\mathbf{h}'}_i &= \tanh(\overrightarrow{\mathbf{W}}[\mathbf{emb}_{x_i}; (\overrightarrow{\mathbf{r}}_i \odot \overrightarrow{\mathbf{h}}_{i-1})]) && (4)\\ \overrightarrow{\mathbf{r}}_i &= \sigma(\overrightarrow{\mathbf{W}}_r[\mathbf{emb}_{x_i}; \overrightarrow{\mathbf{h}}_{i-1}]) && (5) \end{aligned}$$

where $\odot$ denotes element-wise product between vectors and $\mathbf{emb}_{x_i}$ is the word embedding of the x_i. $\tanh(\cdot)$ and $\sigma(\cdot)$ are the tanh and sigmoid transformation functions that can be applied element-wise on vectors, respectively. For simplicity, we

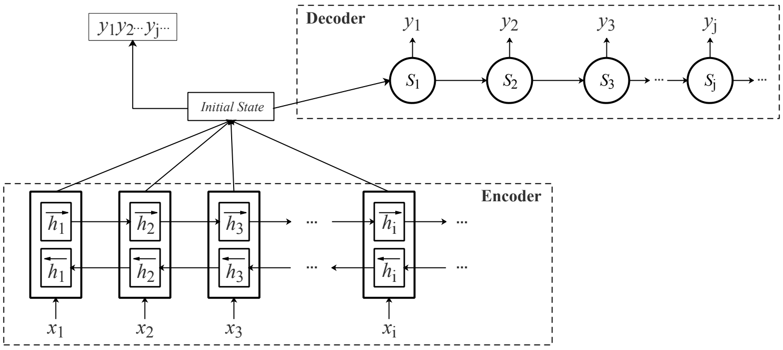

Figure 1: The NMT model with word prediction for the initial state.

omit the bias term in each network layer. The backward RNN encoder is defined likewise.

In the decoding stage, the decoder starts with the initial state $\mathbf{s}_0$, which is the average of source representations (Bahdanau et al., 2014).

$$\mathbf{s}_0 = \sigma(\mathbf{W}_s \frac{1}{|\mathbf{x}|} \sum_{i=1}^{|\mathbf{x}|} \mathbf{h}_i) \tag{6}$$

At each time step j, the decoder maximizes the conditional probability of generating the jth target word, which is defined as:

$$P(y_j|y_{<j}, \mathbf{x}) = f_d(t_d([\mathbf{emb}_{y_{j-1}}; \mathbf{s}_j; \mathbf{c}_j])) \tag{7}$$

$$f_d(\mathbf{u}) = \text{softmax}(\mathbf{W}_f \mathbf{u}) \tag{8}$$

$$t_d(\mathbf{v}) = \tanh(\mathbf{W}_t \mathbf{v}) \tag{9}$$

where $\mathbf{s}_j$ is the decoder's hidden state, which is computed by another GRU (as in Equation 2):

$$\mathbf{s}_j = g_d(\mathbf{s}_{j-1}, [\mathbf{emb}_{y_{j-1}}; \mathbf{c}_j]) \tag{10}$$

and the context vector $\mathbf{c}_j$ is from the attention mechanism (Luong et al., 2015b):

$$\mathbf{c}_j = \sum_{i=1}^{|\mathbf{x}|} a_{ji} \mathbf{h}_i \tag{11}$$

$$a_{ji} = \frac{\exp(e_{ji})}{\sum_{k=1}^{|\mathbf{x}|} \exp(e_{jk})} \tag{12}$$

$$e_{ji} = \tanh(\mathbf{W}_{att_d}[\mathbf{s}_{j-1}; \mathbf{h}_i]). \tag{13}$$

4 NMT with Word Predictions

4.1 Word Prediction for the Initial State

The decoder starts the generation of target sentence from the initial state $\mathbf{s}_0$ (Equation 6) generated by the encoder. Currently, the update for the encoder only happens when a translation error occurs in the decoder. The error is propagated through multiple time steps in the recurrent structure until it reaches the encoder. As there are hundreds of millions of parameters in the NMT system, it is hard for the model to learn the exact representation of source sentences. As a result, the values of initial state may not be exact during the translation process, leading to poor translation performances.

We propose word prediction as a mechanism to control the values of initial state. The intuition is that since the initial state is responsible for the translation of whole target sentence, it should at least contain information of each word in the target sentence. Thus, we optimize the initial state by making prediction for all target words. For simplicity, we assume each target word is independent of each other.

Here the word prediction mechanism is a simpler sub-task of translation, where the order of words is not considered. The prediction task could be trained jointly with the translation task in a multi-task learning way (Luong et al., 2015a; Dong et al., 2015; Zhang and Zong, 2016), where both tasks share the same encoder. In other words, word prediction for the initial state could be interpreted as an improvement for the encoder. We denote this mechanism as WP_E .

As shown in Figure 1, a prediction network is added to the initial state. We define the conditional probability of WP_E as follows:

$$P_{\text{WP}_\text{E}}(\mathbf{y}|\mathbf{x}) = \prod_{j=1}^{|\mathbf{y}|} P_{\text{WP}_\text{E}}(y_j|\mathbf{x}) \tag{14}$$

$$P_{\text{WP}_\text{E}}(y_j|\mathbf{x}) = f_p(t_p([\mathbf{s}_0; \mathbf{c}_p])) \tag{15}$$

where $f_p(\cdot)$ and $t_p(\cdot)$ are the softmax layer and non-linear layer as defined in Equation 8-9, with different parameters; $\mathbf{c}_p$ is defined similar as the

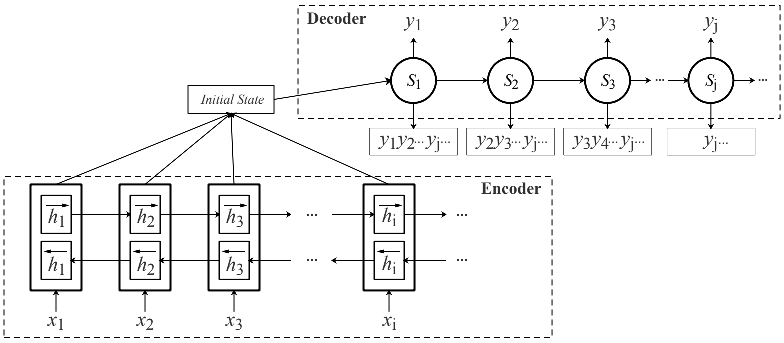

Figure 2: The NMT model with word predictions for the decoder's hidden states.

attention network, so the source side information could be enhanced.

$$\mathbf{c}_p = \sum_{i=1}^{|\mathbf{x}|} a_i \mathbf{h}_i \tag{16}$$

$$a_i = \frac{\exp(e_i)}{\sum_{k=1}^{|\mathbf{x}|} \exp(e_k)} \tag{17}$$

$$e_i = \tanh(\mathbf{W}_{att_p}[\mathbf{s}_0, \mathbf{h}_i]). \tag{18}$$

4.2 Word Predictions for Decoder's Hidden States

Similar intuition is also applied for the decoder. Because the hidden states of the decoder are responsible for the translation of target words, they should be able to predict the target words as well. The only difference is that we remove the already generated words from the prediction task. So each hidden state in the decoder is required to predict the target words which remain untranslated.

For the first state $\mathbf{s}_1$ of the decoder, the prediction task is similar with the task for the initial state. Since then, the prediction is no longer a separate training task, but integrated into each time step of the training process. We denote this mechanism as WP_D.

As shown in Figure 2, for each time step j in the decoder, the hidden state $\mathbf{s}_j$ is used for the prediction of $(y_j, y_{j+1}, \cdots, y_{|\mathbf{y}|})$. The conditional probability of WP_D is defined as:

$$\begin{aligned} P_{\text{WP}_\text{D}}&(y_j, y_{j+1}, \cdots, y_{|\mathbf{y}|}|y_{<j}, \mathbf{x}) \\ &= \prod_{k=j}^{|\mathbf{y}|} P_{\text{WP}_\text{D}}(y_k|y_{<j}, \mathbf{x}) \end{aligned} \tag{19}$$

$$P_{\text{WP}_\text{D}}(y_k|y_{<j}, \mathbf{x}) = f_d(p(t_d([\mathbf{emb}_{y_{j-1}}; \mathbf{s}_j; \mathbf{c}_j]))) \tag{20}$$

where $f_d(\cdot)$ and $t_d(\cdot)$ are the softmax layer and non-linear layer as defined in Equation 8-9; $p(\cdot)$ is another non-linear transformation layer, which prepares the current state for the prediction:

$$p(\mathbf{u}) = \tanh(\mathbf{W}_p \mathbf{u}). \tag{21}$$

4.3 Training

NMT models optimize the networks by maximizing the likelihood of the target translation $\mathbf{y}$ given source sentence $\mathbf{x}$, denoted by L_T.

$$L_\text{T} = \frac{1}{|\mathbf{y}|} \sum_{j=1}^{|\mathbf{y}|} \log P(y_j|y_{<j}, \mathbf{x}) \tag{22}$$

where $P(y_j|y_{<j}, \mathbf{x})$ is defined in Equation 7.

To optimize the word prediction mechanism, we propose to add extra likelihood functions L_{WP_E} and L_{WP_D} into the training procedure.

For the WP_E, we directly optimize the likelihood of translation and word prediction:

$$L_1 = L_\text{T} + L_{\text{WP}_\text{E}} \tag{23}$$

$$L_{\text{WP}_\text{E}} = \log P_{\text{WP}_\text{E}} \tag{24}$$

where P_{WP_E} is defined in Equation 14.

For the WP_D, we optimize the likelihood as:

$$L_2 = L_\text{T} + L_{\text{WP}_\text{D}} \tag{25}$$

$$L_{\text{WP}_\text{D}} = \sum_{j=1}^{|\mathbf{y}|} \frac{1}{|\mathbf{y}| - j + 1} \log P_{\text{WP}_\text{D}} \tag{26}$$

where P_{WP_D} is defined in Equation 19; the coefficient of the logarithm is used to calculate the average probability of each prediction.

The two mechanisms could also work together, so that both the encoder and the decoder could be improved:

$$L_3 = L_\text{T} + L_{\text{WP}_\text{E}} + L_{\text{WP}_\text{D}}. \tag{27}$$

4.4 Making Use of the Word Predictor

The previously proposed word prediction mechanism could be used only as a extra training objective, which will not be computed during the translation. Thus the computational complexity of our models for translation stays exactly the same.

On the other hand, using a smaller and specific vocabulary for each sentence or batch will improve translation efficiency. If the vocabulary is accurate enough, there is also a chance to improve the translation quality (Jean et al., 2015; Mi et al., 2016; L'Hostis et al., 2016). Our word prediction mechanism $\mathrm{WP_E}$ provides a natural solution for generating a possible set of target words at sentence level. The prediction could be made from the initial state $\mathbf{s}_0$, without using extra resources such as word dictionaries, extracted phrases or frequent word lists, as in Mi et al. (2016).

5 Experiments

5.1 Data

We perform experiments on the Chinese-English (CH-EN) and German-English (DE-EN) machine translation tasks. For the CH-EN, the training data consists of about 8 million sentence pairs [1]. We use NIST MT02 as our validation set, and the NIST MT03, MT04 and MT05 as our test sets. These sets have 878, 919, 1597 and 1082 source sentences, respectively, with 4 references for each sentence. For the DE-EN, the experiments trained on the standard benchmark WMT14, and it has about 4.5 million sentence pairs. We use newstest 2013 (NST13) as validation set, and newstest 2014(NST14) as test set. These sets have 3000 and 2737 source sentences, respectively, with 1 reference for each sentence. Sentences were encoded using byte-pair encoding (BPE) (Britz et al., 2017).

5.2 Systems and Techniques

We implement a baseline system with the bi-directional encoder (Bahdanau et al., 2014) and the attention mechanism (Luong et al., 2015b) as described in Section 3, denoted as baseNMT. Then our proposed word prediction mechanism on initial state and hidden states of decoder are implemented on the baseNMT system, denoted as $\mathrm{WP_E}$ and $\mathrm{WP_D}$, respectively. We denote the system use both techniques as $\mathrm{WP_{ED}}$. We implement systems with variable-sized vocabulary following (Mi et al., 2016). For comparison, we also implement systems with dropout (with dropout rate 0.5 on the output layer) and ensemble (ensemble of 4 systems at the output layer) techniques.

5.3 Implementation Details

Both our CH-EN and DE-EN experiments are implemented on the open source toolkit dl4mt [2], with most default parameter settings kept the same. We train the NMT systems with the sentences of length up to 50 words. The source and target vocabularies are limited to the most frequent 30K words for both Chinese and English, respectively, with the out-of-vocabulary words mapped to a special token UNK.

The dimension of word embedding is set to 512 and the size of the hidden layer is 1024. The recurrent weight matrices are initialized as random orthogonal matrices, and all the bias vectors as zero. Other parameters are initialized by sampling from the Gaussian distribution $\mathcal{N}(0, 0.01)$.

We use the mini-batch stochastic gradient descent (SGD) approach to update the parameters, with a batch size of 32. The learning rate is controlled by AdaDelta (Zeiler, 2012).

For efficient training of our system, we adopt a simple pre-train strategy. Firstly, the baseNMT system is trained. The training results are used as the initial parameters for pre-training our proposed models with word predictions.

For decoding during test time, we simply decode until the end-of-sentence symbol eos occurs, using a beam search with a beam width of 5.

5.4 Translation Experiments

To see the effect of word predictions in translation, we evaluate these systems in case-insensitive IBM-BLEU (Papineni et al., 2002) on both CH-EN and DE-EN tasks.

The detailed results are show in the Table 1 and Table 2. Compared to the baseNMT system, all of our models achieve significant improvements. On the CH-EN experiments, simply adding word predictions to the initial state ($\mathrm{WP_E}$) already brings considerable improvements. The average improvement on test set is 2.53 BLEU, showing that constraining the initial state does lead to a higher translation quality. Adding word predic-

[1] includes LDC2002E18, LDC2003E07, LDC2003E14, LDC2004E12, LDC2004T08, LDC2005T06, LDC2005T10, LDC2006E26 and LDC2007T09

[2] https://github.com/nyu-dl/dl4mt-tutorial

Models	MT02(dev)	MT03	MT04	MT05	Test Ave.	IMP
baseNMT	34.04	34.92	36.08	33.88	34.96	–
WP_E	39.36	37.17	39.11	36.20	37.49	**+2.53**
WP_D	40.28	38.45	40.99	37.90	39.11	**+4.15**
WP_ED	40.25	39.50	40.91	38.05	39.49	**+4.53**

Table 1: Case-insensitive 4-gram BLEU scores of baseNMT, WP_E, WP_D, WP_ED systems on the CH-EN experiments. (The "IMP" column presents the improvement of test average compared to the baseNMT.)

Models	NST13(dev)	NST14	IMP
baseNMT	23.56	20.68	–
WP_E	24.44	21.09	**+0.41**
WP_D	25.31	21.54	**+0.86**
WP_ED	25.97	21.98	**+1.3**

Table 2: Case-insensitive 4-gram BLEU scores of baseNMT, WP_E, WP_D, WP_ED systems on the DE-EN experiments.

Models	Test	IMP
baseNMT	34.86	–
WP_ED	39.49	+4.53
baseNMT-dropout	37.02	+2.06
WP_ED-dropout	39.25	+4.29
baseNMT-ensemble(4)	37.71	+2.75
WP_ED-ensemble(4)	40.75	+5.79

Table 3: Average case-insensitive 4-gram BLEU scores on the CH-EN experiments for baseNMT and WP_ED systems, with the dropout and ensemble techniques.

tions to the hidden states in the decoder (WP_D) leads to further improvements against baseNMT (4.15 BLEU), because WP_D adds constraints to the state transitions through different time steps in the decoder. Using both techniques improves the baseline by 4.53 BLEU. On the DE-EN experiments, the improvement of WP_E model is 0.41 BLEU and WP_D model is 0.86 BLEU on test set. When use both techniques, the WP_ED improves on the test set is 1.3 BLEU.

We compare our models with systems using dropout and ensemble techniques. The results show in Table 3 and 4. On the CH-EN experiments, the dropout method successfully improves the baseNMT system by 2.06 BLEU. However, it does not work on our WP_ED system. The ensemble technique improves the baseNMT system by 2.75 BLEU. It still improves WP_ED by 1.26 BLEU, but the improvement is smaller than on the baseNMT. On the DE-EN experiments, the phenomenon of experiments is similar to CH-EN experiments. The baseNMT system improves 0.94 through dropout method and 0.9 BLEU through ensemble method. The dropout technique also does not work on WP_ED and the ensemble technique improves 1.79 BLEU. These comparisons suggests that our system already learns better and stable values for the parameters, enjoying some of the benefits of general training techniques like dropout and ensemble. Compared to dropout and ensemble, our method WP_ED achieves the highest improvement against the baseline system on both CH-EN and DE-EN experiments. Along with ensemble method, the improvement could be up to 5.79 BLEU and 1.79 BLEU respectively.

Models	Test	IMP
baseNMT	20.68	–
WP_ED	21.98	+1.3
baseNMT-dropout	21.62	+0.94
WP_ED-dropout	21.71	+1.03
baseNMT-ensemble(4)	21.58	+0.9
WP_ED-ensemble(4)	22.47	+1.79

Table 4: Case-insensitive 4-gram BLEU scores on the DE-EN experiments for baseNMT and WP_ED systems, with the dropout and ensemble techniques.

5.5 Word Prediction Experiments

Since we include an explicit word prediction mechanism during the training of NMT systems, we also evaluate the prediction performance on the CH-EN experiments to see how the training is improved.

For each sentence in the test set, we use the initial state of the given model to make prediction about the possible words. We denote the set of top n words as T_n, the set of words in all the references

top-n	baseNMT		WP$_{\mathrm{E}}$	
	Prec.	Recall	Prec.	Recall
top-10	45%	17%	73%	30%
top-20	33%	21%	63%	43%
top-50	21%	30%	41%	55%
top-100	14%	39%	28%	68%
top-1k	2%	67%	4%	89%
top-5k	0.7%	84%	0.9%	95%
top-10k	0.4%	90%	0.5%	97%

Table 5: Comparison between baseNMT and WP$_{\mathrm{E}}$ in precision and recall for the different prediction size on the CH-EN experiments.

as R. We define the precision, recall of the word prediction as follows:

$$\text{precision} = \frac{|T_n \cap R|}{|T_n|} * 100\% \tag{28}$$

$$\text{recall} = \frac{|T_n \cap R|}{|R|} * 100\% \tag{29}$$

We compare the prediction performance of baseNMT and WP$_{\mathrm{E}}$. WP$_{\mathrm{ED}}$ has similar prediction results with WP$_{\mathrm{E}}$, so we omit its results. As shown in Table 5, baseNMT system has a relatively lower prediction precision, for example, 45% in top 10 prediction. With an explicit training, the WP$_{\mathrm{E}}$ could achieve a much higher precision in all conditions. Specifically, the precision reaches 73% in top 10. This indicates that the initial state in WP$_{\mathrm{E}}$ contains more specific information about the prediction of the target words, which may be a step towards better semantic representation, and leads to better translation quality.

Because the total words in the references are limited (around 50), the precision goes down, as expected, when a larger prediction set is considered. On the other hand, the recall of WP$_{\mathrm{E}}$ is also much higher than baseNMT. When given 1k predictions, WP$_{\mathrm{E}}$ could successfully predict 89% of the words in the reference. The recall goes up to 95% with 5k predictions, which is only 1/6 of the current vocabulary.

To analyze the process of word prediction, we draw the attention heatmap (Equation 16) between the initial state $\mathbf{s}_0$ and the bi-directional representation of each source side word $\mathbf{h}_i$ for an example sentence. As shown in Figure 3, both examples show that the initial states have a very strong attention with all the content words in the source sentence. The blank cells are mostly functions words or high frequent tokens such as "的 ('s)", "是 (is)", "而 (and)", "它 (it)", comma and period. This indicates that the initial state successfully encodes information about most of the content words in the source sentence, which contributes for a high prediction performance and leads to better translation.

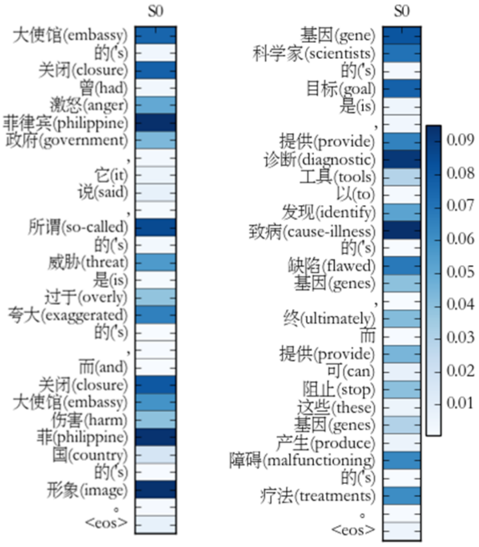

Figure 3: Two examples of the attention heatmap between the initial state $\mathbf{s}_0$ and the bi-directional representation of each source side word $\mathbf{h}_i$ from the CH-EN test sets. (The English translation of each source word is annotated in the parentheses after it.)

5.6 Improving Decoding Efficiency

To make use of the word prediction, we conduct experiments using the predicted vocabulary, with different vocabulary size (1k to 10k) on the CH-EN experiments, denoted as WP$_{\mathrm{E}}$-V and WP$_{\mathrm{ED}}$-V. The comparison is made in both translation quality and decoding time. As all our models with fixed vocabulary size have exactly the same number of parameters for decoding (extra mechanism is used only for training), we only plot the decoding time of the WP$_{\mathrm{ED}}$ for comparison. Figure 4 and 5 show the results.

When we start the experiments with top 1k vocabulary (1/30 of the baseline settings), the translation quality of both WP$_{\mathrm{E}}$-V and WP$_{\mathrm{ED}}$-V are already higher than the baseNMT; while their decoding time is less than 1/3 of an NMT system with 30k vocabulary. When the size of vocabulary increases, the translation quality improves as well. With a 6k predicted vocabulary (1/5 of the baseline settings), the decoding time is about 60% of a full-

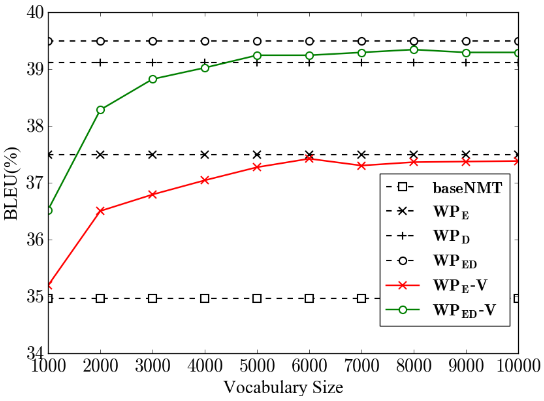

Figure 4: BLEU scores with different vocabulary sizes for each sentence on the CH-EN experiments. (The performance of baseNMT, WP_E, WP_D, WP_{ED} are plotted as horizontal lines for comparison.)

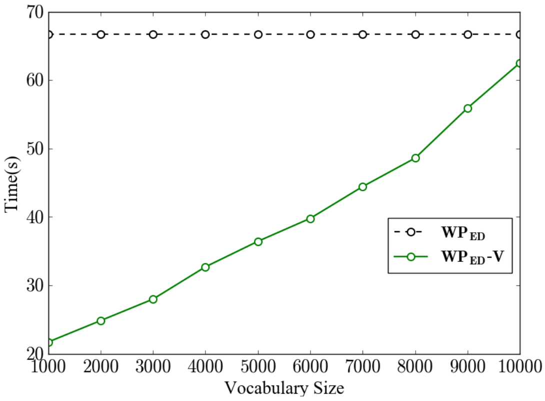

Figure 5: Decoding time with different vocabulary sizes for each sentence on the CH-EN experiments. (The horizontal line shows the decoding time for the systems with fixed vocabulary.)

vocabulary system; the performances of both systems with variable size vocabulary are comparable their corresponding fixed-vocabulary systems, which is higher than the baseNMT by 2.53 and 4.53 BLEU, respectively.

Although the comparison may not be fair enough due to the language pair and training conditions, the above relative improvements (e.g. WP_{ED}-V v.s. baseNMT) is much higher than previous research of manipulating the vocabularies (Jean et al., 2015; Mi et al., 2016; L'Hostis et al., 2016). This is because our mechanism is not only about reducing the vocabulary itself for each sentence or batch, it also brings improvement to the overall translation model. Please note that unlike these research, we keep the target vocabulary to be 30k in all our experiments, because we are not focusing on increasing the vocabulary size in this paper. It will be interesting to combine our mechanism with larger vocabulary to further enhance the translation performance. Again, our mechanism requires no extra annotation, dictionary, alignment or separate discriminative predictor, etc.

5.7 Translation Analysis

We also analyze real-case translations to see the difference between different systems (Table 6).

It is easy to see that the baseNMT system misses the translations of several important words, such as "advertising", "1.5", which are marked with underline in the reference. It also wrongly translates the company name "time warner inc." as the redundant information "internet company"; "america online" as "us line".

The results of dropout or ensemble show improvement compared to the baseNMT. But they still make mistakes about the translation of "online" and the company name "time warner inc.".

With WP_{ED}, most of these errors no longer exist, because we force the encoder and decoder to carry the exact information during translation.

6 Conclusions

The encoder-decoder architecture provides a general paradigm for learning machine translation from the source language to the target language. However, due to the large amount of parameters and relatively small training data set, the end-to-end learning of an NMT model may not be able to learn the best solution. We argue that at least part of the problem is caused by the long error backpropagation pipeline of the recurrent structures in multiple time steps, which provides no direct control of the information carried by the hidden states in both the encoder and decoder.

Instead of looking for other annotated data, we notice that the words in the target language sentence could be viewed as a natural annotation. We propose to use the word prediction mechanism to enhance the initial state generated by the encoder and extend the mechanism to control the hidden states of decoder as well. Experiments show promising results on the Chinese-English and German-English translation tasks. As a by-product, the word predictor could be used to improve the efficiency of decoding, which may be

source	时代华纳公司的网络公司美国线上说, 它预期二○○二年的广告与商业销售将由二○○一年的二十七亿美元减少到十五亿美元。
reference	america online , the internet arm of time warner conglomerate , said it expects advertising and commerce revenue to decline from us $ 2.7 billion in 2001 to us $ 1.5 in 2002 .
baseNMT	in the *us line* , *the internet company 's internet company said on the internet* that it expected that the business sales in 2002 would fall from $ UNK billion to $ UNK billion in 2001 .
baseNMT +dropout	on *the united states line* , *UNK 's* internet company said *on the internet* that it expects to reduce the annual **advertising and commercial** sales from $ UNK billion in 2001 to $ **1.5** billion .
baseNMT +ensemble	in the *us line* , *the internet company 's internet company* said that it expected that the **advertising and commercial** sales volume for 2002 would be reduced from us $ UNK billion to us $ **1.5** billion in 2001 .
WP$_{\text{ED}}$	**the internet company** *of* **time warner inc.** , the *us* online , said that it expects that **the advertising and commercial** sales in 2002 will decrease from $ UNK billion in 2001 to us $ **1.5** billion .

Table 6: Comparisons of different systems in translating the same example sentence, which from CH-EN test sets. ("source" indicates the source sentence; "reference" indicates the human translation; the translation results are indicated by their system names, including our best "WP$_{\text{ED}}$" systems. The underline words in the reference are missed in the baseNMT output; the bold font indicates improvements over the baseNMT system; and the italic font indicates remaining translation errors.)

crucial for large scale applications.

Our attempts demonstrate that the learning of the large scale neural network systems is still not good enough. In the future, it might be helpful to analyze the benefits of jointly learning other related tasks together with machine translation, to provide further control of the learning process. It is interesting to demonstrate the effectiveness of the proposed mechanism on other sequence to sequence learning tasks as well.

Acknowledgments

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by the National Science Foundation of China (No. 61672277, 61472183), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR* abs/1409.0473. http://arxiv.org/abs/1409.0473.

Srinivas Bangalore, Patrick Haffner, and Stephan Kanthak. 2007. Statistical machine translation through global lexical selection and sentence reconstruction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*. Association for Computational Linguistics, pages 152–159. http://aclweb.org/anthology/P07-1020.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive Exploration of Neural Machine Translation Architectures. *ArXiv e-prints* .

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)*. Association for Computational Linguistics, pages 1724–1734. https://doi.org/10.3115/v1/D14-1179.

Junyoung Chung, Çaglar Gülçehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR* abs/1412.3555. http://arxiv.org/abs/1412.3555.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1723–1732. https://doi.org/10.3115/v1/P15-1166.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, pages 1–10. https://doi.org/10.3115/v1/P15-1001.

Minwoo Jeong, Kristina Toutanova, Hisami Suzuki, and Chris Quirk. 2010. A discriminative lexicon model for complex morphology. In *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas (AMTA 2010)*.

Gurvan L'Hostis, David Grangier, and Michael Auli. 2016. Vocabulary selection strategies for neural machine translation. *CoRR* abs/1610.00072. http://arxiv.org/abs/1610.00072.

Minh-Thang Luong, Quoc V. Le, Ilya Sutskever, Oriol Vinyals, and Lukasz Kaiser. 2015a. Multi-task sequence to sequence learning. *CoRR* abs/1511.06114. http://arxiv.org/abs/1511.06114.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. *CoRR* abs/1604.00788. http://arxiv.org/abs/1604.00788.

Minh-Thang Luong, Hieu Pham, and Christopher D. Manning. 2015b. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1412–1421. https://doi.org/10.18653/v1/D15-1166.

Arne Mauser, Saša Hasan, and Hermann Ney. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 210–218. http://aclweb.org/anthology/D09-1022.

Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. *CoRR* abs/1610.05011. http://arxiv.org/abs/1610.05011.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Vocabulary manipulation for neural machine translation. *CoRR* abs/1605.03209. http://arxiv.org/abs/1605.03209.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '02, pages 311–318. https://doi.org/10.3115/1073083.1073135.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* 15(1):1929–1958. http://dl.acm.org/citation.cfm?id=2627435.2670313.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, Curran Associates, Inc., pages 3104–3112. http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks.pdf.

Jun Suzuki and Masaaki Nagata. 2017. Rnn-based encoder-decoder approach with word frequency estimation. *CoRR* abs/1701.00138. http://arxiv.org/abs/1701.00138.

Ke Tran, Arianna Bisazza, and Christof Monz. 2014. Word translation prediction for morphologically rich languages with bilingual neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, pages 1676–1688. https://doi.org/10.3115/v1/D14-1175.

Matthew D. Zeiler. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701. http://arxiv.org/abs/1212.5701.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, pages 1535–1545. http://aclweb.org/anthology/D16-1160.