Implicit Distortion and Fertility Models for Attention-based Encoder-Decoder NMT Model

Shi Feng[†]
Shanghai Jiao Tong University
Shanghai, P.R. China
`sjtufs@gmail.com`

Shujie Liu, Mu Li, Ming Zhou
Microsoft Research
Beijing, P.R. China
`shujliu,muli,mingzhou@microsoft.com`

Abstract

Neural machine translation has shown very promising results lately. Most NMT models follow the encoder-decoder framework. To make encoder-decoder models more flexible, attention mechanism was introduced to machine translation and also other tasks like speech recognition and image captioning. We observe that the quality of translation by attention-based encoder-decoder can be significantly damaged when the alignment is incorrect. We attribute these problems to the lack of distortion and fertility models. Aiming to resolve these problems, we propose new variations of attention-based encoder-decoder and compare them with other models on machine translation. Our proposed method achieved an improvement of 2 BLEU points over the original attention-based encoder-decoder.

1 Introduction

Neural machine translation has shown promising results lately. Most NMT methods follow the encoder-decoder framework proposed by (Cho et al., 2014), which typically consists of two RNNs: the encoder RNN reads the source sentence and transform it into vector representation; the decoder RNN takes the vector representation and generates the target sentence word by word. The decoder will stop once a special symbol denoting the end of the sentence is generated. This encoder-decoder framework can be used on general sequence-to-sequence tasks (Sutskever et al.,

2014), like question answering and text summarization. After some modification, for example replacing the RNN encoder with a CNN, the model can also be applied to tasks like image captioning (Vinyals et al., 2014; Xu et al., 2015). In the following discussion, we focus on the task of machine translation.

In the original encoder-decoder model, although the encoder RNN generates a set of hidden states, one at each position of the source sentence, the decoder only takes the last one. This design in effect compresses the variable-length source sentence into a fixed-length context vector, with the information of each source word implicitly stored in the context vector. Thus the decoder cannot easily make full use of the whole sequence of encoder hidden states. To make it more flexible and generalize the fixed-length representation to a variable-length one, it was proposed to use attention mechanism for machine translation (Bahdanau et al., 2014).

Attention mechanism was first proposed to allow models to learn alignments between different modalities, e.g., between image objects and agent actions in the dynamic control problem (Mnih et al., 2014).

In (Bahdanau et al., 2014), attention mechanism was applied to machine translation to learn an alignment between source words and target words. Fig. 1 shows a sample alignment given by attention mechanism.

With the ability of learning alignments between different modalities from attention mechanism, attention-based encoder-decoder model is more powerful than just encoder-decoder and has been used for many tasks like question answering (Hermann et al., 2015), speech recognition (Bahdanau

---

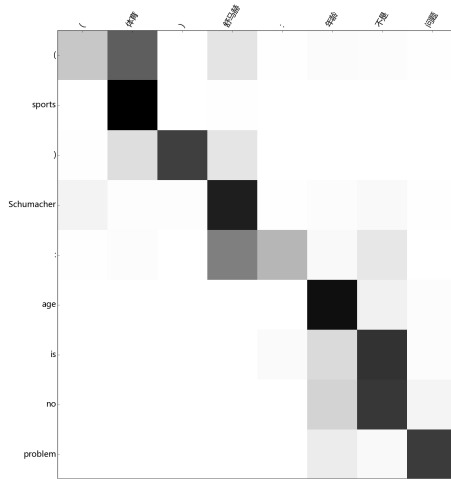[†]Work done while Shi was an intern at Microsoft Research.

Figure 1: Alignment by attention mechanism. Each row is a distribution of how the corresponding target word (English) is aligned to source word. Darker color denotes higher weight.

Figure 2: Wrong alignment due to lack of distortion model.

et al., 2015; Chorowski et al., 2014), image captioning (Xu et al., 2015) and visual question answering (Xu and Saenko, 2015; Chen et al., 2015; Shih et al., 2015). In these applications, variations of attention mechanism were proposed to enhance its performance.

2 Problems of Attention Mechanism

By training the encoder-decoder model with attention mechanism, we get an alignment from target word to source word. This alignment helps translation by allowing re-ordering. But since the alignment by attention is not always accurate, we observed that in many cases where the alignment is incorrect, the translation quality is significantly damaged. We attribute this kind of problem to the lack of explicit distortion and fertility models in attention-based encoder-decoder model.

2.1 Lack of Distortion Model

In SMT, the distortion model controls how the words are re-ordered. In Fig. 2 we show an example alignment given by attention mechanism where incorrect alignment in the middle of the sentence caused the translation to go wrong afterwards. We focus on the later part of the sentence where the correct translation should be:

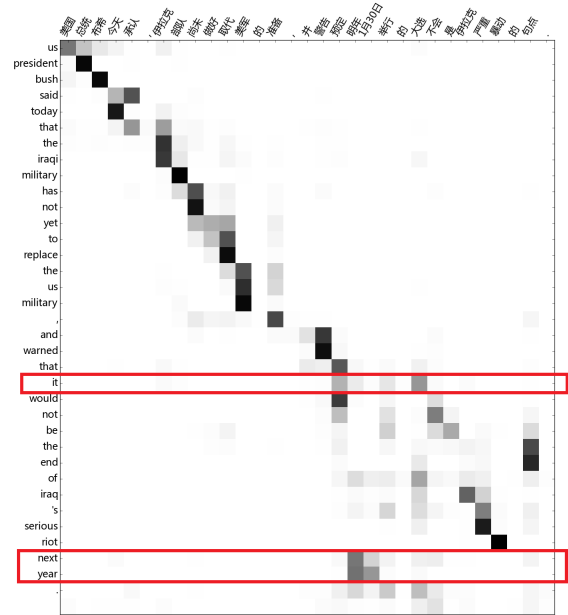"*...and warned that the election to be held on january 30th next year would not be and end to*

serious violence in iraq."

which get translated into:

"*...and warned that it would not be the end of iraq's serious violence* **next year**."

From the alignment matrix we can see that, the word "预定" (means "scheduled") in source sentence is attended to by "*would*", but the next attention jumped to "大选 (means "election")", while it should focus on "明年" (means "next year") or on the date. After this incorrect re-ordering, the meaning of the source sentence is twisted in the translation. If the attention is aware of the previous alignment, it should be able to order "*next year*" after "*election*" and reserve the mearning of source sentence correctly.

In this kind of casese, the translation can go wrong due to incorrect re-ordering. We attribute this kind of problem due to the lack of distortion model in attention-based encoder-decoder.

2.2 Lack of Fertility Model

In SMT, the fertility model controls how many target words are translated from a source word. We observe two phenomena related to the lack of fertility model in attention-based encoder-decoder: the problem of repetition and the problem of coverage.
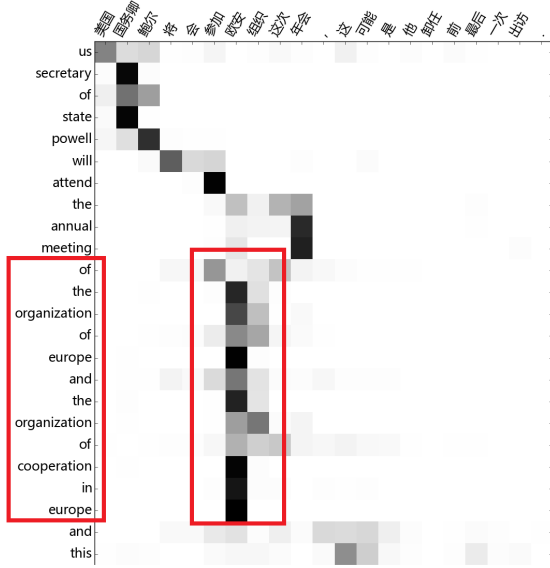
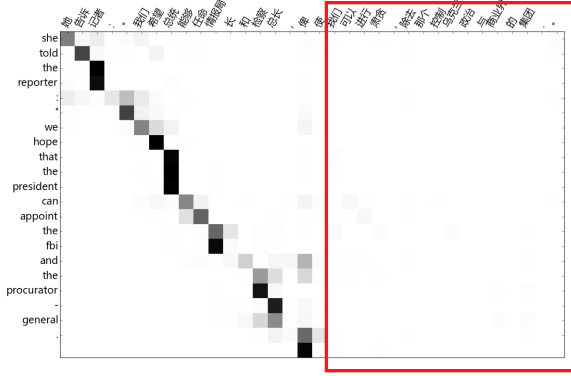Figure 3: Problem of repetition in alignment.

Figure 4: Problem of coverage in alignment.

Problem of Repetition Fig. 3 shows an example of repetition problem in alignment. For consecutive words, the attention mechanism focused on the same position in the source sentence, resulting in repetition in the translation, "*the organization of europe and the organization of cooperation in europe*".

Problem of Coverage Fig. 4 shows an example of coverage problem in alignment. We see that some part of the source sentence was not attended to, resulting in significant loss of content in the translation.

These two problems are due to the lack of fertility model in NMT: in the first case, some source words are translated into too many target words, while in the second case, some sources words are translted into too few target words.

Although attention mechanism already makes

the encoder-decoder more flexible by allowing re-ordering, the observed problems demonstrated some restrictions of it. Motivated by these observations, we propose additions of implicit distortion and fertility models to attention-based encoder decoder. In Sec. 4, we introduce RECATT and RNNATT which are designed as an implicit distortion models. In Sec. 5, we introduce COND-DEC which is designed as an implicit fertility model. We verify that the proposed methods can resolve the observed problems in our experiments in Sec. 8.2.

3 Attention-based Encoder-Decoder

We start by reviewing the RNN used in NMT papers and the encoder-decoder with attention mechanism from (Bahdanau et al., 2014).

3.1 Gated Recurrent Unit

Gated Recurrent Unit (GRU) (Cho et al., 2014) is an RNN alternative similar to LSTM (Hochreiter and Schmidhuber, 1997). It was used in NMT papers (Cho et al., 2014; Bahdanau et al., 2014) and we will use GRU as RNN in our paper. Like normal RNN, GRU computes its hidden state h_i based on the input x_i and previous hidden state h_{i-1}:

$$h_i = \text{RNN}(h_{i-1}, x_i)$$

which is computed with update gate and reset gate, formally defined by:

$$r_i = \sigma(W^r x_i + U^r h_{i-1})$$
$$h'_i = \tanh(r_i \circ U h_{i-1} + W x_i)$$
$$z_i = \sigma(W^z x_i + U^z h_{i-1})$$
$$h_i = (1 - z_i) \circ h'_i + z_i \circ h_{i-1}$$

where x_i is the input, h_{i-1} is the previous hidden state. z_i and r_i are the values of update gate and reset gate respectively. $\circ$ denotes bit-wise product. Biases are dropped for simplicity.

3.2 RNNSearch (Bahdanau et al., 2014)

Encoder The encoder used in RNNSEARCH (Bahdanau et al., 2014) is a bi-directional RNN. It consists of two independent RNNs, one reading the source sentence from left to right, another from right to left:

$$\overrightarrow{s}_i = \text{RNN}(\overrightarrow{s}_{i-1}, x_i)$$
$$\overleftarrow{s}_i = \text{RNN}(\overleftarrow{s}_{i+1}, x_i)$$

where x_i is the word embedding of source word at position i. The representation at position i is then defined as the concatenation of $\overrightarrow{s}_i$ and $\overleftarrow{s}_i$:

$$s_i = \left[\begin{array}{c} \overrightarrow{s}_i \\ \overleftarrow{s}_i \end{array} \right]$$

Decoder with Attention Unlike the decoder from (Cho et al., 2014) which takes only the last representation, the decoder with attention mechanism can make full use of the whole representation set s_j. The decoder is assisted by a unit that provides a dynamic context c_i:

$$c_i = \text{ATT}(h_{i-1}, \{s_j\})$$

At each decoder step, the attention unit takes both the previous decoder hidden state h_{i-1} and the set of encoder representations $\{s_j\}$ as input, outputs a weighted average of encoder hidden states as the context c_i. It uses a match function α to match h_{i-1} with each s_j and generates the weight w_{ij} for s_j.

$$e_{ij} = v^T \tanh \alpha(h_{i-1}, s_j)$$
$$w_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$
$$c_i = \sum_j w_{ij} s_j$$

The match function can take on many forms, which is analyzed in (Luong et al., 2015).

In our paper we use the sum match function, which is a more common choice as used in (Bahdanau et al., 2014).

$$\alpha_{\text{sum}}(a, b) = W^\alpha a + U^\alpha b$$

The context c_i is then used by the decoder:

$$h_i = \text{RNN}(h_{i-1}, y_{i-1}, c_i)$$

where y_{i-1} is the embedding of previous target word.

To predict a target word at position i, the decoder state h_i is concatenated with c_i and y_{i-1} and fed through deep out (Pascanu et al., 2012) with a single maxout hidden layer (Goodfellow et al., 2013), followed by a softmax. We follow this structure in this paper.
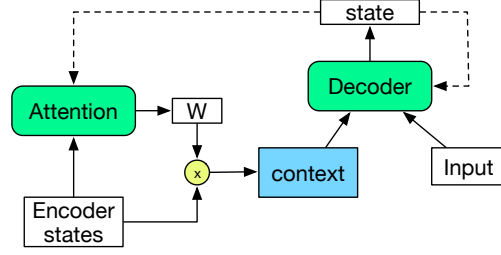

Figure 5: Attention Decoder. The dashed lines show how the current hidden state is passed to the next decoding step. The current attention is computed with the previous hidden state.
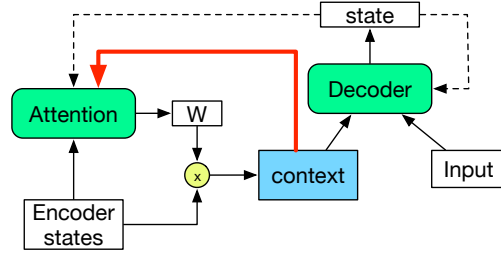

Figure 6: RecAtt Decoder. The red thick line denotes the recurrent attention path which passes previous attention generated context to the attention unit.

4   Recurrent Attention Mechanisms

In Fig. 5 we show an abstraction of the decoder-attention structure.

We note that attention mechanism treats the encoder states as a set, not a sequence, while the source sentence order is crucial to re-ordering. And the state of re-ordering given to the attention unit is all embedded in the hidden state of the decoder - the attention unit itself does not have memory.

Motivated by the analysis in Sec. 2.1, we propose to add recurrent paths to the decoder-attention structure to provide the attention unit with more information the re-ordering. With these recurrent paths, instead of making the decoder remembering what the state of re-ordering is, recurrent attention mechanism explicitly keeps track of this information.

4.1   RecAtt

In this section we introduce our proposed recurrent attention mechanism, RecAtt.

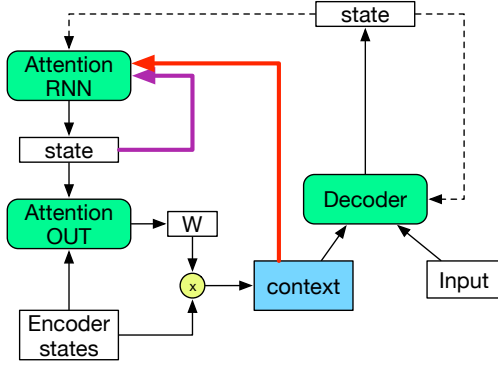We pass the previous context directly to the at-

Figure 7: RNNATT Decoder. The purple thick line denotes the recurrent path of the hidden state of the attention unit. The red thick line denotes passing previous attention generated context to the attention unit for it to update its hidden state with respect to what information was extracted from the source in the previous step.

tention unit to inform it about the alignment from the previous step.

The decoder with RecAtt follows:

$$c_i = \text{ATT}(h_{i-1}, c_{i-1}\{s_j\})$$
$$h_i = \text{RNN}(h_{i-1}, y_{i-1}, c_i)$$

Where the modified attention mechanism RecAtt follows:

$$e_{ij} = v^T \tanh \alpha(h_{i-1}, c_{i-1}, s_j)$$
$$w_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$
$$c_i = \sum_j w_{ij} s_j$$

The modified sum match function is:

$$\alpha_{\text{sum}}(a, b, c) = W^\alpha a + U^\alpha b + V^\alpha c$$

We note that RecAtt is purely content-based - the recurrent attention information is the context vector instead of the weights. We show in our experiments that making attention unit itself recurrent is very important to improving end-to-end translation performance.

4.2 RNNAtt

RecAtt designed with the aim of adding a distortion model. In RecAtt, only the previous attention-generated context is used in the recurrent path, so it only has a "short-term memory". To make it

more flexible and have a longer memory, we propose RNNATT, as shown in Fig. 7. The attention unit now keeps a hidden state and in effect becomes a complete RNN.

$$c_i = \text{ATT\_OUT}(q_{i-1}, \{s_j\})$$
$$q_i = \text{ATT\_RNN}(q_{i-1}, h_{i-1}, c_i)$$
$$h_i = \text{RNN}(y_{i-1}, h_{i-1}, c_t)$$

where ATT_OUT is the original attention unit which applies the match function and softmax, ATT_RNN denotes the hidden state q_i computation of the attention unit.

5 Conditioned Decoder

As analyzed in Sec. 2.2, attention mechanism might produce incorrect alignment and low-quality translation due to the lack of explicit distortion and fertility models. To address this issue, we propose conditioned decoder, CONDDEC, which uses a condition vector to represent what information has been extracted from the source sentence. This can be seen as an implicit fertility model, the condition vector can keep track of how many target words are translated from each source word. We use a structure similar to (Wen et al., 2015) where a predefined condition is used to guide natural language generation. Different from that method, we use a trainable condition initialized with the last encoder hidden state. At each decoding step, the condition is updated with the decoder state and used to compute the next decoder state. The decoder GRU with attention and condition sd_i is defined as adding an extra decay gate vd_i to decoder:

$$r_i = \sigma(W^r x_i + U^r h_{i-1} + V^r c_i)$$
$$h'_i = \tanh(r_i \circ U h_{i-1} + W x_i + V c_i)$$
$$z_i = \sigma(W^z x_i + U^z h_{i-1} + V^z c_i)$$
$$d_i = \sigma(W^d x_i + U^d h_{i-1} + V^d c_i)$$
$$sd_i = d_i \circ sd_{i-1}$$
$$h_i = (1 - z_i) \circ h'_i + z_i \circ h_{i-1} + \tanh(V^h sd_i)$$

Let T be the length of the source sentence. We further penalize the condition by adding the following two costs to the categorical cross-entropy cost of the translation model:

Step-decay cost We restrict the decay gate from extracting too much information from the condi-
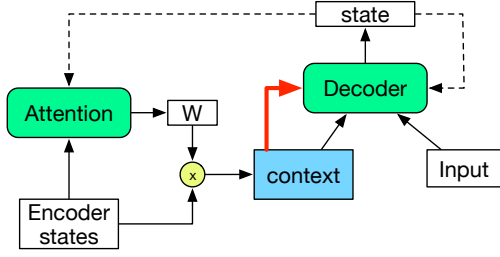
Figure 8: INPUTFEED Decoder. The red thick line denotes the recurrent attention path which passes previous attention generated context to the decoder.
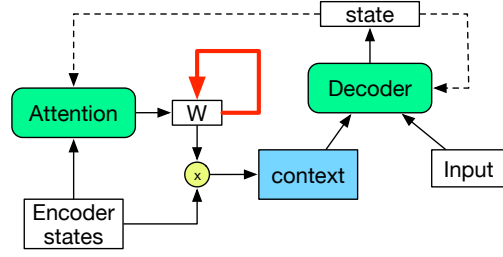

Figure 9: HYBRIDATT1 RNN Decoder. The red thick line denotes the recurrent attention path which uses previous attention weights to adjust current attention weights.

tion. So we add a cost term:

$$\text{cost}_{decay} = \frac{1}{T} \sum_{j=1}^{T} ||\boldsymbol{sd}_j - \boldsymbol{sd}_{j-1}||_2$$

Left-over cost We want the decoder to extract as much information as possible from the condition after reading the source sentence. So we add a cost term:

$$\text{cost}_{left} = ||\boldsymbol{sd}_T||_2$$

These two costs are added to the categorical cross-entropy cost of the translation model. At training time, the costs are used to enforce a fertility model and are ignored at test time.

6   Related work

There are variations of attention mechanism that have recurrent paths similar to that of RECATT. In this section, we review these models and compare those decoder-attention structures.

6.1   InputFeed (Luong et al., 2015)

In (Luong et al., 2015) the authors explored several variations of attention mechanism, including different match functions and local attention. We focus on the input-feeding method proposed in this paper because it is recurrent-like. INPUTFEED passes the previous attention output to the decoder together with current attention output, to further inform the decoder with previous alignment decisions.

$$\boldsymbol{c}_i = \text{ATT}(\boldsymbol{h}_{i-1}, \{\boldsymbol{s}_i\})$$
$$\boldsymbol{h}_t = \text{RNN}(\boldsymbol{h}_{i-1}, \boldsymbol{y}_{i-1}, \boldsymbol{c}_i, \boldsymbol{c}_{i-1})$$

This attention mechanism is purely content-based - the recurrent information is the context given by attention mechanism instead of weights. Note that the recurrent information is used outside the attention function, directly to the decoder, which makes it different from RecAtt, where the recurrent information is passed to the attention unit.

6.2   HybridAtt1(Chorowski et al., 2014)

In (Chorowski et al., 2014) the authors proposed an attention mechanism with a recurrent path. When computing the current set of weights on encoder states, the attention unit takes the previous weights and penalize the jump distance. It computes the average attention center, which is $m_{i-1} = \sum_j j * w_{(i-1),j}$. Then it adjusts the weight of each encoder state by its distance from that center.

$$m_{i-1} = \sum_j j \cdot w_{i-1,j}$$
$$e_{ij} = \boldsymbol{v}^T \tanh \alpha(\boldsymbol{h}_{i-1}, \boldsymbol{s}_j)$$
$$e'_{ij} = \text{Logistic}(j - m_{i-1}) \cdot \exp(e_{ij})$$
$$w_{ij} = \frac{e'_{ij}}{\sum_k e'_{ik}}$$
$$\boldsymbol{c}_i = \sum_j w_{ij} \boldsymbol{s}_j$$

This is a content-based attention with location-based recurrent attention, which is characterized by using an average attention center. Note that the recurrent information is used outside the attention unit, to adjusting the weights, which makes it different from RecAtt.
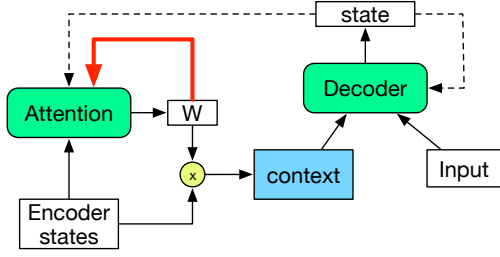
Figure 10: HYBRIDATT2 RNN Decoder. The red thick line denotes the recurrent attention path which passes previous attention weights convoluted with $\boldsymbol{Q}$ to the attention unit.

6.3 HybridAtt2(Bahdanau et al., 2015)

In paper (Bahdanau et al., 2015) the authors followed the previous one and improved HYBRIDATT1 by integrating the recurrent location information into attention function. It first extracts feature vectors $\boldsymbol{g}_i$ by doing convolution with previous weights $\boldsymbol{Q} * w_{i-1}$, then uses these feature vectors to predict new weights.

$$\boldsymbol{g}_i = \boldsymbol{Q} * \boldsymbol{w}_{i-1}$$
$$e_{ij} = \boldsymbol{v}^T \tanh \alpha(\boldsymbol{h}_{i-1}, s_j, \boldsymbol{g}_{ij})$$
$$w_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$
$$\boldsymbol{c}_i = \sum_j w_{ij} \boldsymbol{s}_j$$

where $*$ denotes convolution.

This is also a content-based attention mechanism with location-based recurrent attention. The difference between this method and HYBRIDATT1 is that the recurrent information is integrated into the attention function.

We note that HYBRIDATT1 and HYBRIDATT2 were proposed for speech recognition task, which requires less re-ordering compared to translation. Although these models improved the performance of encoder-decoder on speech recognition, they not necessarily will on machine translation. We included these models because they have structures similar to RECATT.

Other variations of attention mechanism with similar recurrent paths include (Mnih et al., 2014), (Chen et al., 2015). In these works, the authors used attention mechanism on image classification and visual question answering respectively. The variations of attention mechanism they used are location-based attention, which is more reasonable

for image-related tasks. Due to these reasons we do not review or compare their methods in this work.

7 Experimental Setup

In this section we describe the data used in our experiments, our evaluation methods and our validation procedure.

Datasets For training, we use NIST Chinese-English training set excluding the Hong Kong Law and Hong Kong Hansard (0.5m sentence pairs after exclusion). For testing, we use Nist2005 dataset (1082 sentence pairs). For validation, we use Nist2003 dataset (913 sentence pairs). Validation set is only used for early-stopping and training process monitoring.

Following (Bahdanau et al., 2014), we use source and target dictionaries of size 30000, covering 97.4% and 98.9% of the vocabularies respectively. Out-of-vocabulary words are replaced with a special token ⟨UNK⟩.

Post-processing We perform post-processing based on the alignment given by attention mechanism. For each translated target word, we choose the source word assigned with the highest attention weight as the aligned word.

⟨UNK⟩'s in the translated sentence are replaced with the correct translation of the aligned source word. We make a simple word-level translation table from the alignment result given by GIZA++ (Och and Ney, 2003) from the training set: for each source word, we choose the most frequently aligned target word.

Evaluation Performance is evaluated by BLEU score (Papineni et al., 2002) over the test set.

We compare 6 models, RNNSEARCH (Cho et al., 2014), HYBRIDATT2 (Bahdanau et al., 2015), IN-PUTFEED (Luong et al., 2015), and three proposed models, RECATT, RNNATT and COND-DEC. We skip HYBRIDATT1 because we have HYBRIDATT2 as an improved version.

We benchmark the 6 NMT models with our implementation of hierarchical phrase-based SMT from (Chiang, 2007), with standard features, denoted as SMT.

Validation Validation is done by calculating the BLEU score over the validation set without

	Before	After	Improvement
SMT	/	32.25	/
RNNSEARCH	26.65	31.02	4.37
HYBRIDATT2	24.60	29.14	4.54
INPUTFEED	25.44	29.02	3.58
RECATT	**28.10**	**33.14**	**5.04**
RNNATT	25.04	30.02	4.98
CONDDEC	27.48	32.21	4.73

Table 1: BLEU scores w/o post-processing and the improvement from post-processing

Figure 11: Effect of implicit distortion model.

post-processing, using `MultiBleu.perl` script from (Bahdanau et al., 2014). For each model, we choose the parameters of the highest validation score.

Model Training The encoder and decoder have 1000 hidden units each. The dimension of source and target word embedding is 620. Following (Bahdanau et al., 2014), we use dropout rate 0.5.

We remove sentences of length over 50 words from the training set. We use batch size of 80 with 12 batches pre-fetched and sorted by the sentence length.

Each model is trained with AdaGrad (Duchi et al., 2011) on K40m GPU for approximately 4 days, finishing over 400000 updates, equivalent to 640 epochs.

When testing trained models, we use beam search (Graves, 2012; Boulanger-Lewandowski et al., 2013; Sutskever et al., 2014) with beam size of 12.

8 Results

8.1 Quantitative

BLEU scores on the test set are shown in (Table 1).

RECATT performed best among NMT models, with and without post-processing. RECATT achieved a 2.1 BLEU score improvement over the original RNNSEARCH.

Note that RECATT also gained the most improvement from post-processing, 5.04 BLEU points. In the post-processing, we use a naive translation table which is generated purely from the training data so the effect of post-processing depends largely on the quality of the alignment. Thus the gain from post-processing can be seen 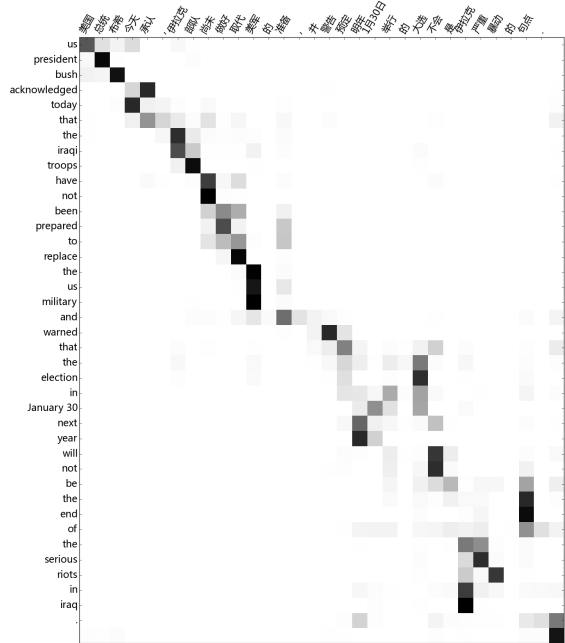as a measurement of the quality of attention-generated alignment, and from this we see that RECATT improved attention mechanism.

CONDDEC out-performed RNNSEARCH by 1 BLEU point, both with and without post-processing.

All three of our proposed models out-performed the phrase-based SMT baseline.

The combination of CONDDEC with RECATT and RNNATT is a work in progress.

8.2 Qualitative

As mentioned in Sec. 2, the original attention-based encoder-decoder has some problems due to the lack of distortion and fertility models. In this section we will qualitatively evaluate how our models resolved these problems.

Distortion We show the alignment and translation by RECATT in Fig. 11 on the same sentence of Fig. 2. In the alignment by RECATT, it can be seen that "*will not*" are correctly aligned to "不会" (means "will not") and "*next year*" is correctly ordered to describe "*the election to be held*" instead of "*riot in iraq*". The transltion quality of the whole sentence is also higher.

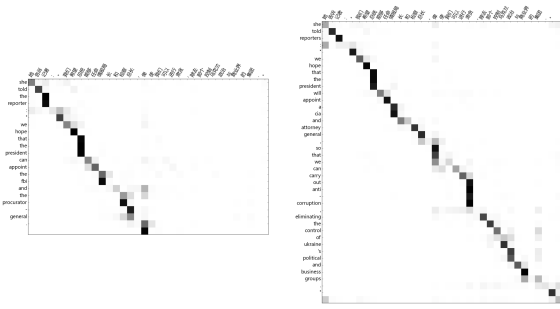**Fertility: Coverage** In Fig. 12 we show the alignments given by RNNSEARCH and RECATT.

Figure 12: Example of coverage problem.
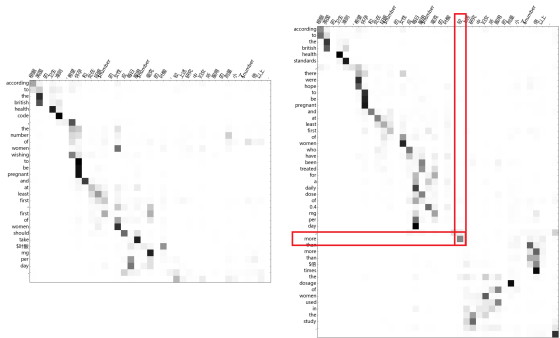Left: RNNSEARCH. Right: CONDDEC.

Figure 14: Example of repetition problem.
Left: RNNSEARCH. Right: CONDDEC.

Figure 13: Example of coverage problem.
Left: RNNSEARCH. Right: RECATT.

Figure 15: Long repetition.

From the alignment of RNNSEARCH, we can observe the problem of coverage where the later part of the source sentence is lost in the translation, while the alignment given by RECATT does not have this problem and covered the whole source sentence.

We observed that RECATT can also resolve the coverage problem. This is because a correct alignment can be very helpful in preventing the incorrect generation of end-of-sentence symbol. In Fig. 13 we show an example. In the alignment by RNNSEARCH, when generating the word after "," (last row), the attention is not very concentrated, leading to the generation of end-of-sentence symbol. While in the alignment by RECATT when generating that word, the attention correctly focused on "较" (means "more") with high confidence, leading to the correct generation of "*more*".

Fertility: Repetition   In Fig. 14 we see that the problem of repetition occurred in the alignment by RNNSEARCH. "东方 快车" (means "midnight express") is repeatedly focused on and translated into "*moon ice*" and "*night of the midnight of*
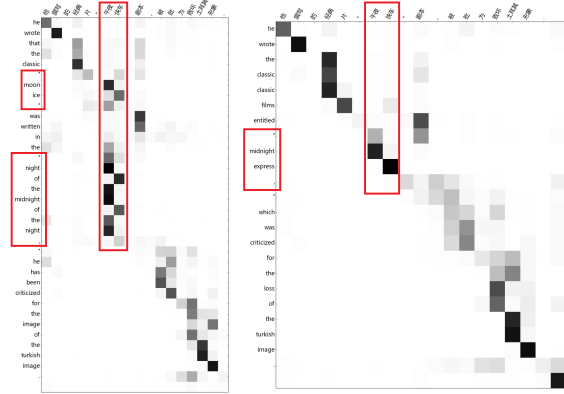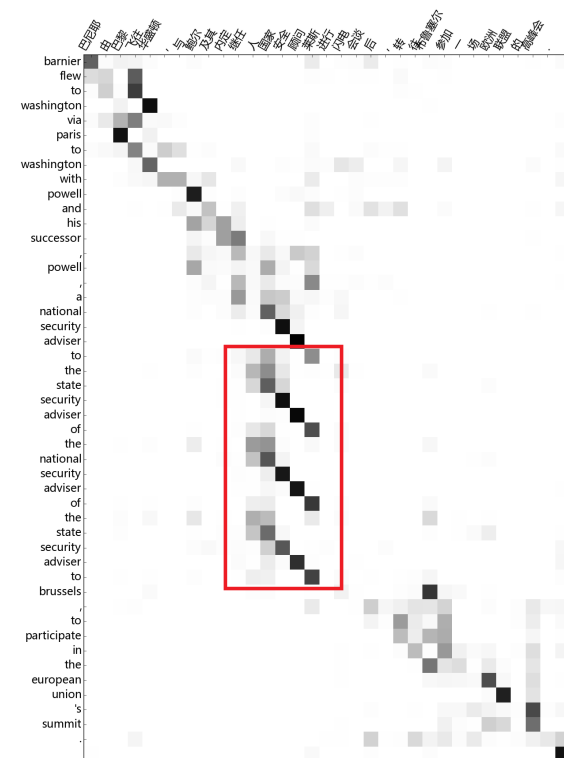
the night". CONDDEC produces both the correct alignment and the correct translation "*midnight express*".

Long Repetition   We observed that RECATT can also resolve the repetition problem. Because the previous attention-generated context was passed to the attention unit, the attention can decide not to focus on the same position as last time. But since it only has a short-term memory, in some cases
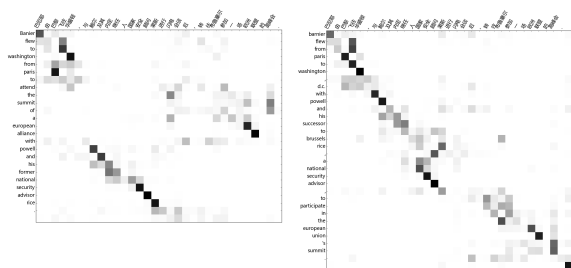
Figure 16: Example of long repetition.
Left: RNNSEARCH. Right: RNNATT.

the alignment by RECATT has long repetitions as shown in Fig. 15.

Although RNNATT did not perform as well as RECATT in terms of BLEU score, we observe that it can resolve the long repetition problem that is hard for RECATT to handle. In Fig. 16 we show the alignments by RNNSEARCH and RNNATT. First we can see that RNNSEARCH did not handle this sentence well, with incorrect alignment and low-quality translation, which shows that this sentence is hard to translate. However the alignment by RNNATT is more accurate and the translation quality is higher than both RNNSEARCH and RE-CATT, with no long repetition problem.

One possible reason for the low BLEU score of RNNATT is that the path from translation cost to the attention recurrent unit is too long and the model is hard to train. Improving end-to-end performance and exploring alternative structures for RNNATT is a work in progress.

9 Conclusions

In this paper we noted some problems occurred in neural machine translation due to the lack of distortion and fertility model.

To resolve these problems, we proposed to add implicit distortion and fertility models to attention-based encoder-decoder. We proposed recurrent attention mechanism, RECATT and RN-NATT for distortion model, and CONDDEC for fertility model. We compared our models with other related variations.

We evaluated our methods both quantitatively and qualitatively. In Chinese-English translation, RECATT gained an improvement of 2 BLEU points over the original attention mechanism and out-performed all related models. CONDDEC also out-performed the original attention mechanism by 1 BLEU point. By analyzing the alignment ma-

trix produced by attention mechanism, we demonstrated that our proposed methods help resolve the observed problems.

We are working on the combination of COND-DEC with RECATT and RNNATT. We will also explore alternative structures of recurrent attention mechanism and try to improve the end-to-end performance of RNNATT.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon Brakel, and Yoshua Bengio. 2015. End-to-end attention-based large vocabulary speech recognition. *arXiv preprint arXiv:1508.04395*.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. 2013. Audio chord recognition with recurrent neural networks. In *ISMIR*, pages 335–340.

Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. 2015. Abc-cnn: An attention based convolutional neural network for visual question answering. *arXiv preprint arXiv:1511.05960*.

David Chiang. 2007. Hierarchical phrase-based translation. *computational linguistics*, 33(2):201–228.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*.

Jan Chorowski, Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. End-to-end continuous speech recognition using attention-based recurrent nn: First results. *arXiv preprint arXiv:1412.1602*.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159.

Ian J Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. 2013. Maxout networks. *arXiv preprint arXiv:1302.4389*.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1684–1692.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.

Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in Neural Information Processing Systems*, pages 2204–2212.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

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*, pages 311–318. Association for Computational Linguistics.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2012. On the difficulty of training recurrent neural networks. *arXiv preprint arXiv:1211.5063*.

Kevin J Shih, Saurabh Singh, and Derek Hoiem. 2015. Where to look: Focus regions for visual question answering. *arXiv preprint arXiv:1511.07394*.

Ilya Sutskever, Oriol Vinyals, and Quoc VV Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2014. Show and tell: A neural image caption generator. *arXiv preprint arXiv:1411.4555*.

Tsung-Hsien Wen, Milica Gasic, Nikola Mrksic, Pei-Hao Su, David Vandyke, and Steve Young. 2015. Semantically conditioned lstm-based natural language generation for spoken dialogue systems. *arXiv preprint arXiv:1508.01745*.

Huijuan Xu and Kate Saenko. 2015. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *arXiv preprint arXiv:1511.05234*.

Kelvin Xu, Jimmy Ba, Ryan Kiros, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*.