Improving Attention Modeling with Implicit Distortion and Fertility for Machine Translation

Shi Feng [1] † **Shujie Liu** [2] **Nan Yang** [2] **Mu Li** [2] **Ming Zhou** [2] **Kenny Q. Zhu** [3]
[1] University of Maryland, College Park
[2] Microsoft Research, Beijing, China
[3] Shanghai Jiao Tong University, Shanghai, China
shifeng@cs.umd.edu
shujliu,nanya,muli,mingzhou@microsoft.com
kzhu@cs.sjtu.edu.cn

Abstract

In neural machine translation, the attention mechanism facilitates the translation process by producing a soft alignment between the source sentence and the target sentence. However, without dedicated distortion and fertility models seen in traditional SMT systems, the learned alignment may not be accurate, which can lead to low translation quality. In this paper, we propose two novel models to improve attention-based neural machine translation. We propose a recurrent attention mechanism as an implicit distortion model, and a fertility conditioned decoder as an implicit fertility model. We conduct experiments on large-scale Chinese–English translation tasks. The results show that our models significantly improve both the alignment and translation quality compared to the original attention mechanism and several other variations.

1 Introduction

Sequence-to-sequence neural machine translation (NMT) has shown promising results lately (Sutskever et al., 2014; Cho et al., 2014b). An NMT model typically consists of an encoding neural network which transforms the source sentence into some vector representation, and a decoding neural network which generates the target sentence from the vector representation. This is called the encoder-decoder model. In order to handle variable length inputs, recurrent neural networks (RNN) are usually used as the encoder and the decoder. The encoder RNN will read the words in the source sentence one by one and generate a sequence of corresponding hidden states; the decoder will then by conditioned on the encoder states to output each word in the target sentence. In (Cho et al., 2014b), only the last encoder state is used for target sentence generation, so the single hidden state vector must preserve all the necessary information in the source sentence for the decoding process , which is very difficult when the source sentence is long.

To leverage the whole sequence of encoder states and retrieve information from the source sentence in a more flexible way, the attention mechanism (Bahdanau et al., 2014) was introduced into the encoder-decoder model. In an attention-based encoder-decoder model, matching scores between the source and target words are calculated based on their corresponding encoder and decoder states. These scores are then normalized and used as weights for the source words given each target word. This can be seen as a soft alignment and the attention mechanism here plays similar role to that of a traditional alignment model.

In alignment models used in traditional machine translation models such as IBM Models (Brown et al., 1993), distortion and fertility are modeled explicitly. By comparison, in the attention mechanism, alignment is computed by matching the previous decoder hidden state with all the encoder hidden states, without modeling distortion and fertility. Since the translation of target words is guided by the attention

† Work done while author was an undergraduate student of Shanghai Jiao Tong University and intern at Microsoft Research.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3082–3092, Osaka, Japan, December 11-17 2016.

mechanism, the translation accuracy of an attention-based NMT model is largely dependent on the accuracy of the alignment, and a large portion of errors seen in the translation result can be associated with the lack of distortion and fertility models. Without a distortion model, the generated alignment sometimes contains incorrect word reordering and as a result the meaning of the sentence could be twisted. Due to the lack of a fertility model, the number of times that each word in the source sentence be aligned to is not restricted, and as a result we sometimes observe that part of the sentence is translated repeatedly, or part of the sentence is missing in the translation.

In the following sections, we first review the attention-based encoder-decoder model, and then give a detailed analysis of these problems using example alignment matrices generated by the standard model. In Section 4 we introduce the two proposed extensions to the attention-based encoder decoder. We first introduce a recurrent attention mechanism with extra recurrent paths as an implicit distortion model to solve the reordering problem. To address the lack of fertility model, we use a fertility vector which memorizes the words that have been translated and design a decoder that is conditioned on this vector. In Section 6 we will show the results of our experiments on large-scale Chinese–English translation tasks and demonstrate that our proposed methods can significantly improve the translation performance.

2 Attention-based Encoder-Decoder

Long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and gated recurrent unit (GRU) (Cho et al., 2014b) are often used as RNNs in attention-based encoder-decoder models. In this section, we will briefly introduce GRU, followed by a short review of how attention is modeled between encoder and decoder states as described in (Bahdanau et al., 2014).

2.1 Gated Recurrent Unit

At time i, a recurrent function RNN computes its hidden state $\boldsymbol{h}_i$ based on the input $\boldsymbol{x}_i$ and previous hidden state $\boldsymbol{h}_{i-1}$:

$$\boldsymbol{h}_i = \text{RNN}(\boldsymbol{h}_{i-1}, \boldsymbol{x}_i)$$

A GRU uses reset gate and update gate to help model long-term dependencies:

$$\begin{aligned}
\boldsymbol{r}_i &= \sigma(\boldsymbol{W}^r \boldsymbol{x}_i + \boldsymbol{U}^r \boldsymbol{h}_{i-1}) \\
\boldsymbol{z}_i &= \sigma(\boldsymbol{W}^z \boldsymbol{x}_i + \boldsymbol{U}^z \boldsymbol{h}_{i-1}) \\
\boldsymbol{h}'_i &= \tanh(\boldsymbol{U}(\boldsymbol{r}_i \odot \boldsymbol{h}_{i-1}) + \boldsymbol{W}\boldsymbol{x}_i) \\
\boldsymbol{h}_i &= (1 - \boldsymbol{z}_i) \odot \boldsymbol{h}'_i + \boldsymbol{z}_i \odot \boldsymbol{h}_{i-1}
\end{aligned}$$

where $\boldsymbol{x}_i$ is the input, and $\boldsymbol{h}_{i-1}$ is the previous hidden state. $\boldsymbol{r}_i$ and $\boldsymbol{z}_i$ are reset and update gates respectively. $\odot$ denotes element-wise product.

2.2 RNNSearch

RNNSEARCH referes to the attention-based encoder-decoder model proposed by (Bahdanau et al., 2014). It consists of two RNNs: an encoder RNN that maps the source sentence to a sequence of hidden states, and a decoder RNN that generates the target sentence based on the encoder states with attention mechanism.

Encoder The encoder used in RNNSEARCH is a bi-directional GRU. It consists of two independent RNNs, one reading the source sentence from left to right, another from right to left, generating two hidden states at each position. The two hidden states produced by forward and backward RNNs are concatenated to generate the sequence of encoder states $\boldsymbol{s}_1^J$, where J is the length of source sentence.

Decoder Unlike the decoder of (Cho et al., 2014b; Sutskever et al., 2014), which takes only the last encoder state as the context vector, the decoder with attention mechanism uses encoder states from all time-stamps as context. Decoder with attention mechanism is illustrated in Figure 5.

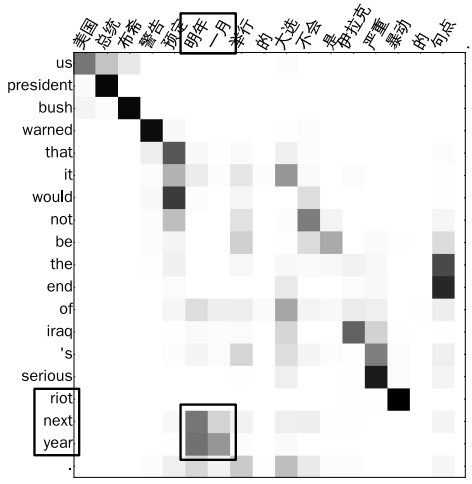

Figure 1. Incorrect reordering by the attention mechanism. The correct translation is "US president Bush warned that the election to be held on January 30th next year would not be an end to serious violence in Iraq."

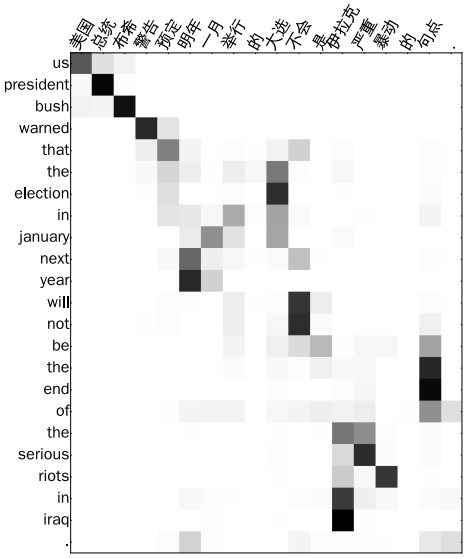

Figure 2. Our proposed model RECATT produced the correct reordering of the source words, and based on that generated a better translation.

At position i in the target sentence, the attention model computes a matching score e_{ij} with match function α, for the previous decoder state $\boldsymbol{h}_{i-1}$ and each encoder state $\boldsymbol{s}_j$.

$$e_{ij} = \boldsymbol{v}^\top \tanh\left(\alpha(\boldsymbol{h}_{i-1}, \boldsymbol{s}_j)\right)$$
$$w_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

We wrap this computation of weights as ALIGN:

$$\boldsymbol{w}_i = \text{ALIGN}(\boldsymbol{h}_{i-1}, \boldsymbol{s}_1^J)$$

There are various match functions, as analyzed in (Luong et al., 2015). In our paper we use the sum match function $\alpha(\boldsymbol{h}_{i-1}, \boldsymbol{s}_j) = \boldsymbol{W}^\alpha \boldsymbol{h}_{i-1} + \boldsymbol{U}^\alpha \boldsymbol{s}_j$. The weighted average of the encoder states $\boldsymbol{s}_1^J$ is calculated as the context $\boldsymbol{c}_i = \sum_j w_{ij} \boldsymbol{s}_j$. It is added to the input of each gate in the decoder, together with previous state $\boldsymbol{h}_{i-1}$ and previous target word embedding $\boldsymbol{y}_{i-1}$:

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

3 Problems of the Attention Mechanism

Although attention modeling works well in finding translation correspondence between source and target words, there are still some issues that can be systematically identified, which fall into three categories: incorrect reordering, missing translation and repeated translation.

Incorrect Reordering Reordering is often required for the translation between two languages with different grammars. When the source words are translated in the wrong order, the meaning of the sentence can be twisted.In the example shown in Figure 1, the phrase "明年 一月" (meaning "January next year") in the source is attended to after the translation of "暴动" (meaning "riot"), resulting in a translation that twisted the meaning of the source sentence.

Missing Translation In Figure 3, we can see that only the first half of the source sentence is translated, because the last half sentence is never chosen for attention.

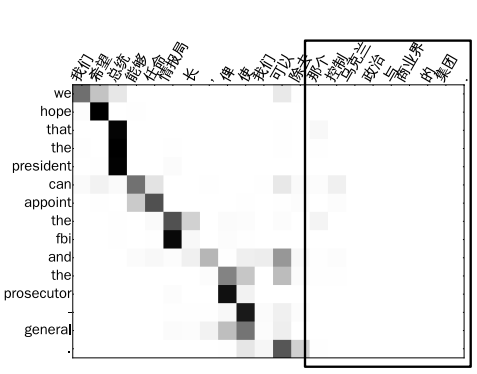

Figure 3. Missing translation example. The correct translation is "We hope that the president could appoint the chief of the intelligence bureau so we can eliminate groups that control Ukrainian's politics and business."

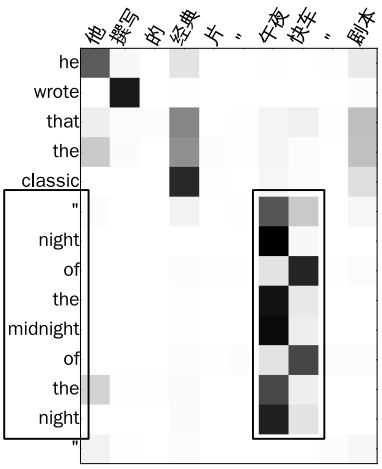

Figure 4. Repeated translation example. The correct translation is "Powell will attend the annual meeting of the organization of europe."

Repeated Translation In the example shown in Figure 4, part of the source sentence, "欧安 组织" ("the organization of europe"), is repeatedly translated into "the organization of europe the organization of europe". This is because the attention mechanism focused on this phrase twice.

4 Our Methods

In traditional SMT methods, the distortion model controls the order of target word generation. It can thus prevent the meaning of source sentences to be twisted due to wrong reordering. We propose to address the incorrect reordering problem using a implicit distortion model which leverages information about previous alignments.

In traditional SMT methods, the fertility model controls how many target words should be generated from a source word. It can thus prevent a source word to be repeatedly translated, which corresponds to the repeated translation problem, or not translated, which corresponds to the missing translation problem. We propose to address the missing and repeated translation problems in NMT by by using a fertility model which memorizes which words have been translated and which have not.

In the following sections, we introduce our extended attention-based encoder-decoder models. For implicit distortion model, we propose a recurrent attention mechanism, RECATT; for implicit fertility model, we propose a fertility-conditioned decoder FERTDEC.

4.1 RECATT

The structure of RECATT is illustrated in Figure 6. At position i in the target sentence, the attention model outputs a weight vector for the encoder states and a weighted-average context. To inform the attention model about the previous alignments, we pass the previous context vector $\boldsymbol{c}_{i-1}$ to it. The decoder with RECATT follows:

$$\boldsymbol{w}_i = \text{ALIGN}(\boldsymbol{h}_{i-1}, \boldsymbol{c}_{i-1}, \boldsymbol{s}_1^J)$$

$$\boldsymbol{c}_i = \sum_{j=1}^{J} w_{ij} \boldsymbol{s}_j$$

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

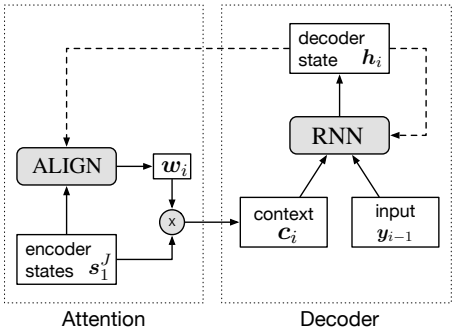

Figure 5. Decoder with attention mechanism. The dashed lines denote passing the previous state to the current attention model and current state.

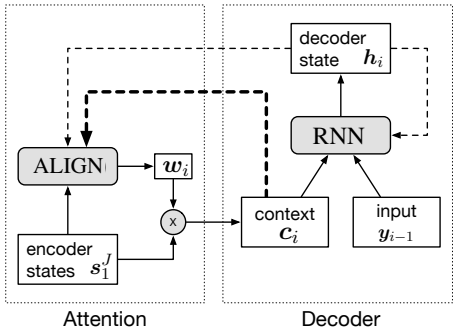

Figure 6. RECATT, decoder with recurrent attention mechanism. The thick dashed line denotes passing the previous attention-generated context to the attention model.

The new ALIGN function with modified match function α is computed as:

$$\alpha(\boldsymbol{h}_{i-1}, \boldsymbol{c}_{i-1}, \boldsymbol{s}_j) = \boldsymbol{W}^\alpha \boldsymbol{h}_{i-1} + \boldsymbol{U}^\alpha \boldsymbol{c}_{i-1} + \boldsymbol{V}^\alpha \boldsymbol{s}_j$$
$$e_{ij} = \boldsymbol{v}^\top \tanh \alpha(\boldsymbol{h}_{i-1}, \boldsymbol{c}_{i-1}, \boldsymbol{s}_j)$$
$$w_{ij} = \frac{\exp(e_{ij})}{\sum_k \exp(e_{ik})}$$

By using the previous context vector, the new attention model can avoid focusing on the same position repeatedly, or jumping from the previous attended position incorrectly. We note that RNNSEARCH is a special case of RECATT where the previous context $\boldsymbol{c}_{i-1}$ is ignored in the match function.

One important design choice of RECATT is to use the previous context vector instead of the previous weight vector. Using the context vector makes the attention model aware of the content of source words, instead of the weight vector, which contains only the position information. Furthermore, the length of the source sentence is variable, so is the length of the weight vector. To use it in the match function, transformation to a fixed-length vector is needed. Possible methods including taking a fixed-size window or passing it through a convolution, both result in a local and partial recurrent information. When we need a long-distance jump from the previous attended position, especially out of the window, partial information might not suffice. Using the context vector, as in RECATT, is not restricted in this scenario. The attention model can always have full information about the previous alignments even when a long-distance jump happens, which makes the implicit distortion model much more flexible.

4.2 FERTDEC

To address the missing and repeated translation problems, we introduce fertility-conditioned decoder FERTDEC. FERTDEC uses a coverage vector [1] to represent the information of the source sentence that has not been translated. Initialized by the sum of source word embeddings $\sum_{j=1}^{J} \boldsymbol{x}_j$ and updated along the translation dynamically, our trainable coverage vector is different from the predefined condition vector used in (Wen et al., 2015). In order to leverage the coverage vector in decoding, we change the

[1]The coverage vector in our work plays a similar role with the one used in beam search decoder (Koehn, 2004).There are two major two differences between them: 1. our coverage vector is used as a soft constraint instead of a hard constraint. 2. we tract untranslated words instead of translated words.

decoding recurrent unit as follows:

$$
\begin{aligned}
\boldsymbol{d}_i &= \boldsymbol{e}_{i-1} \odot \boldsymbol{d}_{i-1} \\
\boldsymbol{r}_i &= \sigma(\boldsymbol{W}^r \boldsymbol{y}_{i-1} + \boldsymbol{U}^r \boldsymbol{h}_{i-1} + \boldsymbol{V}^r \boldsymbol{d}_i) \\
\boldsymbol{z}_i &= \sigma(\boldsymbol{W}^z \boldsymbol{y}_{i-1} + \boldsymbol{U}^z \boldsymbol{h}_{i-1} + \boldsymbol{V}^z \boldsymbol{d}_i) \\
\boldsymbol{e}_i &= \sigma(\boldsymbol{W}^e \boldsymbol{y}_{i-1} + \boldsymbol{U}^e \boldsymbol{h}_{i-1} + \boldsymbol{V}^e \boldsymbol{d}_i) \\
\boldsymbol{h}'_i &= \tanh(\boldsymbol{U}(\boldsymbol{r}_i \odot \boldsymbol{h}_{i-1}) + \boldsymbol{W}\boldsymbol{y}_{i-1} + \boldsymbol{V}\boldsymbol{d}_i) \\
\boldsymbol{h}_i &= (1 - \boldsymbol{z}_i) \odot \boldsymbol{h}'_i + \boldsymbol{z}_i \odot \boldsymbol{h}_{i-1} + \tanh(\boldsymbol{V}^h \boldsymbol{d}_i)
\end{aligned}
$$

where $\boldsymbol{d}_i$ is the coverage vector, $\boldsymbol{e}_i$ is the new added *extract gate*, which is used to update $\boldsymbol{d}_i$ based on the words that has been translated.

$\boldsymbol{d}_i$ is designed to track the untranslated words during decoding, so it is not expected to change drastically between consecutive time-stamps. Also, it should converge to zero at the end of the sentence. Therefore in the training stage, we update the loss function as follows:

$$
\sum_{i=1}^{T} -\log p(y_i) + \frac{1}{T} \sum_{i=1}^{T} ||\boldsymbol{d}_i - \boldsymbol{d}_{i-1}||_2 + ||\boldsymbol{d}_i||_2
$$

where the first term is the negative log-likelihood used in the encoder–decoder model. The new introduced second and third terms are **step-decay** and **left-over** costs. **Step-decay** cost prevents the extract gate from extracting too much information at each time-step. It is different than that of (Wen et al., 2015) [2]. While **left-over** cost ensures all the source words are translated after generating the whole target sentence.

5 Related Work

There are variations of the attention mechanism with recurrent paths similar to those in our recurrent attention mechanism. In this section, we put them in a general framework and compare them with ours.

INPUTFEED Input-feeding method (Luong et al., 2015) also has a recurrent path - the previous attention-generated context is passed to the decoder together with current one:

$$
\begin{aligned}
\boldsymbol{w}_i &= \text{ALIGN}(\boldsymbol{h}_{i-1}, \boldsymbol{s}_1^J) \\
\boldsymbol{c}_i &= \sum_{j=1}^{J} w_{ij} \boldsymbol{s}_j \\
\boldsymbol{h}_i &= \text{RNN}(\boldsymbol{h}_{i-1}, \boldsymbol{y}_{i-1}, \boldsymbol{c}_i, \boldsymbol{c}_{i-1})
\end{aligned}
$$

Using the previous context helps the decoder generate better target words, but it doesn't help the attention model select source words more accurately or generate better alignment. This makes INPUTFEED very different from our RECATT.

MARKOV In Markov condition model (Cohn et al., 2016), ξ takes a fixed-width window of the previous weight vector $\boldsymbol{w}_{i-1}$ and passes it to the attention model:

$$
\begin{aligned}
\xi(\boldsymbol{w}_{i-1}, j) &= [w_{i-1,j-k}, .., w_{i-1,j}, .., w_{i-1,j+k}]^\top \\
\boldsymbol{w}_i &= \text{ALIGN}(\boldsymbol{h}_{i-1}, \boldsymbol{s}_1^J, \xi(\boldsymbol{w}_{i-1})) \\
\boldsymbol{c}_i &= \sum_{j=1}^{J} w_{ij} \boldsymbol{s}_j
\end{aligned}
$$

This can be seen as a location-based counterpart of RECATT. As discussed in Section 4.1, this method is less flexible - it can only use partial recurrent information and is not content-aware.

[2] These two cost functions achieve similar result on our task, but our has no hyperparameter.

LOCFER In local fertility model (Cohn et al., 2016), ξ uses all previous weight vectors $\boldsymbol{w}_{<i}$ and computes the sum of previous attention weights.

$$\xi(\boldsymbol{w}_{<i}, j) = \left[\sum_{i'<i} w_{i',j-k}, .., \sum_{i'<i} w_{i',j+k}\right]^\top$$
$$\boldsymbol{w}_i = \text{ALIGN}(\boldsymbol{h}_{i-1}, \boldsymbol{s}_1^J, \xi(\boldsymbol{w}_{<i}))$$
$$\boldsymbol{c}_i = \sum_{j=1}^{J} w_{ij} \boldsymbol{s}_j$$

The intuition is to consider the fertility up to the current position, and use it to guide new alignment. This is done by a location-based recurrent path similar to that of MARKOV. LOCFER can prevent focusing nearby words already translated, and it is a blend of distortion and fertility model.

6 Experiments

6.1 Settings

Datasets We use NIST Chinese–English training set excluding Hong Kong Law and Hong Kong Hansard as the training set (500,000 sentence pairs after exclusion). The test set is Nist2005 (1082 sentence pairs). The validation set is Nist2003 (913 sentence pairs).

Following (Bahdanau et al., 2014), we use a vocabulary size of 30,000 for both source and target language, covering 97.4% and 98.9% of the words. Out-of-vocabulary words are replaced with a special token $\langle$UNK$\rangle$.

UNK Replacement With word alignment result on the training set generated by GIZA++ (Och and Ney, 2003), we build a translation table. We choose the most frequently aligned target word as the translation for each source word. UNK replacement is performed after the translation is completed, based on the alignment matrix generated by the attention model. If a target word is UNK, we replace it with the translation (from the translation table) of its aligned source word, the one with the highest attention weight.

Model & Baseline Two baseline systems are used in our experiment. The first one is HPSMT, our in-house implementation of hierarchical phrase-based SMT (Chiang, 2007) with standard features. For a fair comparison, the 4-gram language model is trained only with the target sentences of the training set. The second one is RNNSEARCH [3] (Cho et al., 2014b), the original attention-based encoder-decoder. Other compared models are our implementations of: INPUTFEED (Luong et al., 2015), MARKOV and LOCFER (Cohn et al., 2016) as discussed in Section 5.

Training Details For all the NMT models, the hidden GRU states are 1000-dimensional, source and target word embeddings are 620-dimensional. Dropout rate is 0.5. The settings of other hyperparameters follow (Bahdanau et al., 2014). Each model is trained with AdaGrad (Duchi et al., 2011) on a K40m GPU for approximately 4 days, finishing about $400,000$ updates, equivalent to 64 epochs.

6.2 Experiment Results

6.2.1 End-to-end Translation Quality

BLEU scores on the test set are shown in Table 1. The two proposed methods RECATT and FERTDEC both out-performed the original model RNNSEARCH. Note that RECATT gained the most improvement from UNK replacement, 5.04 BLEU points. The effectiveness of our UNK replacement depends largely on the quality of the alignment, so the gain can be seen as a measurement of alignment quality. This is an evidence that RECATT improved attention-generated alignment and as a result improved translation quality. The last line shows the results obtained by the combination of RECATT and FERTDEC, which further out-performed both models.

[3]The implementation of RNNSEARCH is from `https://github.com/mila-udem/blocks-examples`

	Before	After	Diff
HPSMT	/	32.25	/
RNNSEARCH	26.65	31.02	4.37
INPUTFEED	25.44	29.02	3.58
LOCFER	27.05	31.68	4.63
MARKOV	27.54	32.21	4.67
RECATT	28.10	33.14	5.04
FERTDEC	27.51	32.44	4.93
RECATT + FERTDEC	28.87	33.76	4.89

Table 1. BLEU scores w/o UNK replacement and the improvement from UNK replacement.

	SAER	AER
RNNSEARCH	54.75	44.13
RECATT	52.88	42.51
FERTDEC	52.70	42.37
RECATT + FERTDEC	52.40	42.11

Table 2. AER & SAER scores, lower is better.

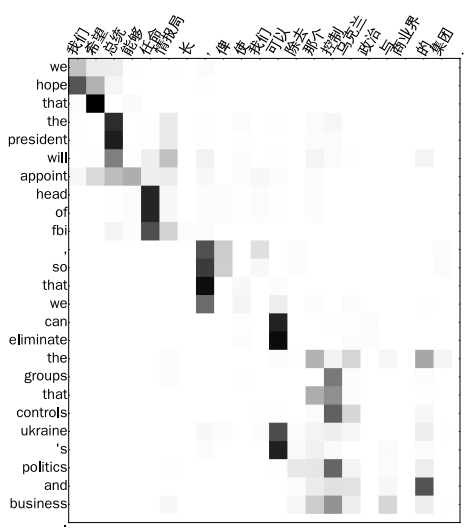

Figure 7. FERTDEC resolved the problem of missing translation problem that is shown in Figure 3.

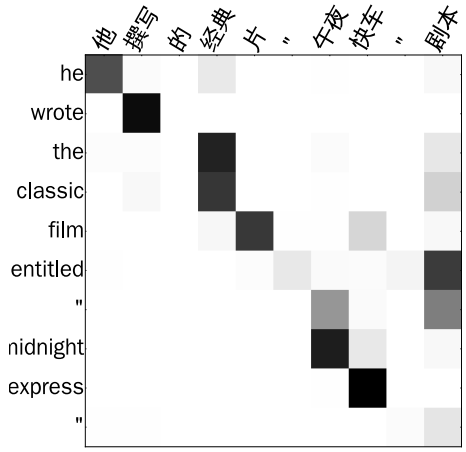

Figure 8. FERTDEC resolved the problem of repeated translation shown in Figure 4.

6.2.2 Alignment Quality

To analyze the effect of our extentions to the attention mechanism in detail, we evaluate the quality of attention-generated alignment by computing the AER (Och and Ney, 2003) and smoothed-AER (Tu et al., 2016) scores on a manually aligned Chinese–English alignment dataset (Haghighi et al., 2009), which contains 491 sentence pairs. We force the model to generate the correct target sentence and evaluate the attention-generated alignment matrix. From the results shown in Table 2, we can see that all three proposed methods achieved better alignment quality, compared with the original attention method.

6.3 Qualitative Analysis

In this section we qualitatively evaluate how our models addressed the problems analyzed in Section 3. All examples shown are from the test set.

Incorrect Reordering In Figure 2 we can see, RECATT generated the correct alignment on the example sentence shown in Figure 1: "will not" is correctly aligned to "不会" (means "will not") and "next year" is correctly translated after "the election to be held" instead of "riot in iraq". The meaning of the source sentence is correctly preserved in the translation.

Missing Translation As shown in Figure 7, FERTDEC resolved the missing translation problem of RNNSEARCH on the same sentence shown in Figure 3. All the information from the source sentence is captured by the translation.

Repeated Translation In Figure 8 we can see that, FERTDEC resolved the the repetition problem of RNNSEARCH shown in Figure 4. "东方 快车" (means "midnight express") is repeatedly focused on and translated into "night of the midnight of the night". As shown on the right, FERTDEC produces both the correct alignment and the correct translation "midnight express".

7 Conclusions and Future Work

In this paper we demonstrated how distortion and fertility models can improve the quality of alignment learned by attention mechanism in encoder-decoder models. We proposed recurrent attention mechanism RECATT as implicit distortion models, and FERTDEC as an implicit fertility model. We conducted various experiments and verified that our proposed methods can improve translation quality by generating better alignment. Compare to the original attention-based encoder-decoder, our best result achieved an improvement of over 2 BLEU points on large-scale Chinese–English translation task.

Our RECATT model is a simple yet effective extension to the attention mechanism, and potentially we can design more complicated mechanisms to model the distortion even better. The key observation is, in RECATT, only the previous context vector is used to provide information about previous alignments, and in effect only the alignment of the previous target word is considered. To extend this short-term information to a long-term one so that the model is aware of all previous alignments, we designed a attention unit that contains a recurrent neural network to encode all previous context vectors. The hidden state vector of this RNN should contain all the information about previous alignments. However in our experiment, this model and several variants did not perform as well as RECATT. But we still think that trying to provide more information about previous alignments, as a natural extension to this work, has the potential of improving both the alignment and translation accuracy.

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*.

Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. 2012. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*.

Yoshua Bengio, Patrice Simard, and Paolo Frasconi. 1994. Learning long-term dependencies with gradient descent is difficult. *Neural Networks, IEEE Transactions on*, 5(2):157–166.

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. 2010. Theano: a cpu and gpu math expression compiler. In *Proceedings of the Python for Scientific Computing Conference (SciPy)*, volume 4, page 3. Austin, TX.

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

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

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, Dzmitry Bahdanau, and Yoshua Bengio. 2014a. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014b. 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*.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. *arXiv preprint arXiv:1601.01085*.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard M Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 1370–1380. Association for Computational Linguistics.

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*.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Aria Haghighi, John Blitzer, John DeNero, and Dan Klein. 2009. Better word alignments with supervised itg models. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 923–931. Association for Computational Linguistics.

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.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, volume 3, page 413.

Philipp Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. In *Machine translation: From real users to research*, pages 115–124. Springer.

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*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685*.

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.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Coverage-based neural machine translation. *arXiv preprint arXiv:1601.04811*.

Bart van Merriënboer, Dzmitry Bahdanau, Vincent Dumoulin, Dmitriy Serdyuk, David Warde-Farley, Jan Chorowski, and Yoshua Bengio. 2015. Blocks and fuel: Frameworks for deep learning. *arXiv preprint arXiv:1506.00619*.

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*.