Deep Attentive Structured Language Model based on LSTM

Di Cao and Kai Yu

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering
Brain Science and Technology Research Center
Shanghai Jiao Tong University, Shanghai, China
{caodi0207,kai.yu}@sjtu.edu.cn

Abstract. Language model (LM) plays an essential role in natural language processing tasks. Given the context, the language model can predict the next word. However, when the history becomes longer, the single hidden vector may be not big enough to store the entire information. In this paper, we propose a deep attentive structured language model (DAS LM), which extends the Long Short-Term Memory (LSTM) neural network with the attention mechanism. With the alternative input of part of speech (POS) tags, the language model is capable of extracting relations between a word and its context. Our model is evaluated on Penn Treebank, Chinese short message and Swb-Fisher corpora. The experiments in language modeling show that our model achieves significant improvements compared to the conventional LSTM language model.

Keywords: language model, long short-term memory, attention mechanism, part of speech

1 Introduction

Language model plays an essential role in natural language processing tasks. Given the word sequence, the language model can compute the probability distribution of the next word. N-gram language model used to be the most popular language model. However, the prediction is only related to the closest several words, the N-gram language model cannot model the long-term dependencies. When the N becomes larger, the sample space will increase exponentially, which induces the problem of data sparseness.

Feedforward Neural Network (FNN) was first introduced into language modeling in 2003 [1]. The neural network based language model has solved some problems such as data sparseness, sample space, and word similarity, but the problem of long-term dependencies still remained unsettled. In 2010, recurrent neural network (RNN) was applied to establish language models, which solved the long-term dependencies problem to a great extent [2] [3]. Currently, Long Short-Term Memory language model (LSTM LM) [4] [5] is widely used for its good performance.

However, there still remains limitations for these RNN based language models. The next word is predicted based on its history, which is compressed and blended into a single dense vector [6], conditionally independent of other states. The relation between the current word and its context is also not known. And, although word embeddings may carry abundant information, they are mainly produced according to the relative positions of words, which may ignore the linguistical information of the sentences.

In this paper, a deep attentive structured language model is proposed. The model is capable of extracting the relation between the hidden state and its input history, applied with the attention module. Using the attention mechanism, we can compute a context input to enhance the language model. Introducing linguistical information to the language model proves to be effective [7]. So, POS tags are applied as an alternative input of our attention based model. The experiments on Penn Treebank, Chinese short messages (SMS) and Swb-Fisher show that our model gains obvious improvements on perplexity (PPL). Interpolated with conventional LSTM LM, it gains significant improvements.

The rest of the paper is organized as follows, section 2 is the background. Section 3 explains the deep attentive structured language model and section 4 shows the experimental setup and results. Finally, conclusion will be given in section 5.

2 Backgourd

2.1 Long Short-Term Memory

RNN is a neural network used to deal with sequential data with its circle structure. Encoding the contextual information into the history hidden, RNN can deal with data of unlimited history length. However, conventional RNN is usually faced with the situations of vanishing or exploding gradients [8], which make it difficult to learn long-term dependency.

LSTM is raised for solving the vanishing and exploding gradient problems. Using three gates, input gate, forget gate and output gate, which control the data flow, LSTM can achieve better performances compared to conventional RNN, and it is widely used in natural language processing.

2.2 External Memory and Attention Mechanism

RNN based models compress and blend history information into a single dense vector [6], conditionally independent of other states. However, when the history becomes longer, the single vector may be not big enough to store the entire information. Using external memory to store the whole history states is one solution [9]. However, approaches to taking full advantage of the external memory remains to study. Attention is a mechanism for reasoning the relation between two vectors [6]. In external memory structure, attention mechanism can also be applied to generate vectors representing the history information [10] [11] [12].

Bahdanau D et al. proposed a general attention mechanism[13]. The main job of the attention model is to score each context vector with respect to the current hidden of the decoder. [11]

2.3 Long Short-Term Memory-Network

Cheng Jianpeng et al. proposed a Long Short-Term Memory-Network (LSTMN) [6], which is a reading simulator that can be used for sequence processing tasks. The LSTMN stores the contextual states in a memory slot and is capable of reasoning about relations between tokens with a neural attention layer [6]. The hidden states and memory states of the LSTM are stored in two tapes, H_{t-1} and C_{t-1}. The attention layer computes the relation between x_t and h_i according to the formulas below.

$$e_i^t = v^T Tanh(W_h h_i + W_x x_t + W_{\tilde{h}} \tilde{h}_{t-1}) \tag{1}$$

$$a_i^t = Softmax(e_i^t) \tag{2}$$

where W_* are weight parameters, and v is a vector parameter. a_i^t is a probability distribution over the hidden state vectors of previous tokens [6].

$$\begin{bmatrix} \tilde{h}_t \\ \tilde{c}_t \end{bmatrix} = \sum_{i=1}^{t-1} a_i^t \cdot \begin{bmatrix} h_i \\ c_i \end{bmatrix} \tag{3}$$

$\tilde{h}_t$ and $\tilde{c}_t$ can be computed respectively by the weighted sum of h_i and c_i. Replacing the h_t, c_t by $\tilde{h}_t$, $\tilde{c}_t$, the next hidden vector h_{t+1} and memory vector c_{t+1} can be computed by the LSTM cell.

$$\begin{bmatrix} h_{t+1} \\ c_{t+1} \end{bmatrix} = LSTM(x_t, \begin{bmatrix} \tilde{h}_t \\ \tilde{c}_t \end{bmatrix}) \tag{4}$$

3 Deep Attentive Structured Language Model

3.1 Attention Mechanism

Conventional RNN based language model predicts the next word based on the context information stored in the hidden layer, which is a single dense vector. When the hisotry becomes longer, the single vector may be not big enough to store the entire information. In the paper, a deep attentive structured language model (DAS LM) is proposed which considers long-term contexts. The model utilizes external memory to store entire inputs. An attention mechanism is applied to produce the context vector as the additional input, to enhance the LSTM. The network of the DAS LM is shown in Fig. 1.

In the DAS language model, the module to compute the context vector w'_t is called the controller module. According to the input history w_i and the previous hidden state h_{t-1}, the controller module is able to extract their relation and compute the corresponding context vector w'_t.

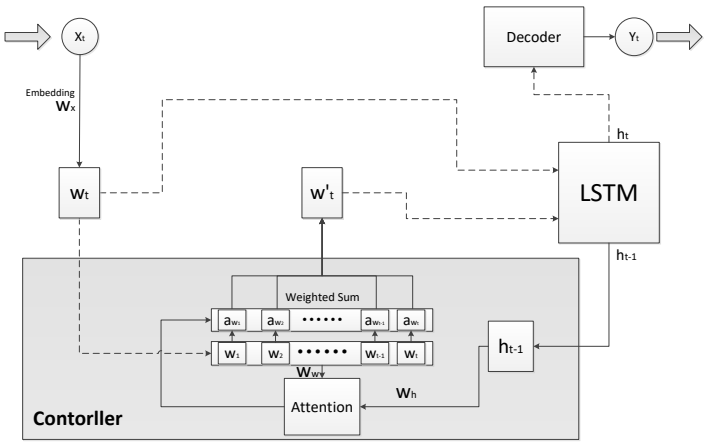

Fig. 1. The network of the deep attentive structured language model. The dashed arrows indicate that the connections are applied with dropout operation.

The relation between the previous hidden state h_{t-1} and the input history $w_1, w_2 ... w_{t-1}, w_t$ can be scored by the following formulas.

$$w_t = W_x x_t \tag{5}$$

$$e_i^t = v^T Tanh(W_w w_i + W_h h_{t-1} + b_{att}) + b_v \tag{6}$$

where W_x, W_w, W_h are weight parameters, v^T is a vector parameter, b_{att}, b_v are bias parameters, and e_i^t denotes the unnormalized relation weight between h_{t-1} and w_i.

Softmax operation is applied to normalize the weights. The values of the weights indicate the relation between the previous hidden state and the input history. Then we can obtain the context vector w_t' by calculating the weighted sum of the input history.

$$a_i^t = \frac{exp(e_i^t)}{\sum_{j=1}^{t} exp(e_j^t)} \tag{7}$$

$$w_t' = \sum_{i=1}^{t} a_i^t \cdot w_i \tag{8}$$

Feeding w_t and w_t' into the LSTM cell, we can compute the hidden and the memory vectors h_t, c_t.

$$f_t = \sigma(W_f[h_{t-1}, w_t, w_t'] + b_f) \tag{9}$$

$$i_t = \sigma(W_i[h_{t-1}, w_t, w_t'] + b_i) \tag{10}$$

$$o_t = \sigma(W_o[h_{t-1}, w_t, w'_t] + b_o) \tag{11}$$

$$\hat{c}_t = tanh(W_{\hat{c}}[h_{t-1}, w_t, w'_t] + b_{\hat{c}}) \tag{12}$$

$$c_t = f_t * c_{t-1} + i_t * \hat{c}_t \tag{13}$$

$$h_t = o_t * tanh(c_t) \tag{14}$$

Finally, the probability distribution of the predicted word y_t can be calculated.

$$y_t = Softmax(W_y h_t + b_y) \tag{15}$$

where W_* are weight parameters, b_* are bias parameters and σ is the sigmoid function.

3.2 Attention Mechanism with a POS Tagging Model

Although word embeddings carry abundant information, they are mainly produced according to the relative positions of words, which may ignore the linguistical information of the sentences. In our model, linguistical information such as part of speech (POS) can be applied as an alternative input to promote the language model.

POS Tagging Model In traditional grammar, a part of speech is a category of words (or, more generally, of lexical items) which have similar grammatical properties [14]. POS tagging is a classic classification task. Given the word sequence w_1, w_2, ..., w_n, the tagging model predicts the aligned tag sequence $y_1, y_2, ..., y_n$. Mainstream tagging models [15] utilize the contextual information of the input word sequence, which means the former tags may be decided by latter words. However, for most language modeling task, future information shouldn't be provided, which means bidirectional tagging models may not be taken into consideration. In this case, an unidirectional LSTM based model is established for POS tagging.

Combination with a POS Tagging Model The network of the DAS LM with a POS Tagging model is shown in fig. 2. x_t and l_t denote the one-hot representations of the word and its POS tag at time step t. They are embedded separately and passed to controller modules.

$$w_t = W_x x_t \tag{16}$$

$$p_t = W_l l_t \tag{17}$$

where W_x and W_l are weight parameters.

Applying the the attention mechanism in section 3.1, we can calculate both the context vectors w'_t and p'_t of the word history and the tag history respectively. Concatenating the current inputs w_t and p_t, w'_t and p'_t respectively, we can obtain the combined input m_t and context vector m'_t.

$$m_t = [w_t, p_t] \tag{18}$$

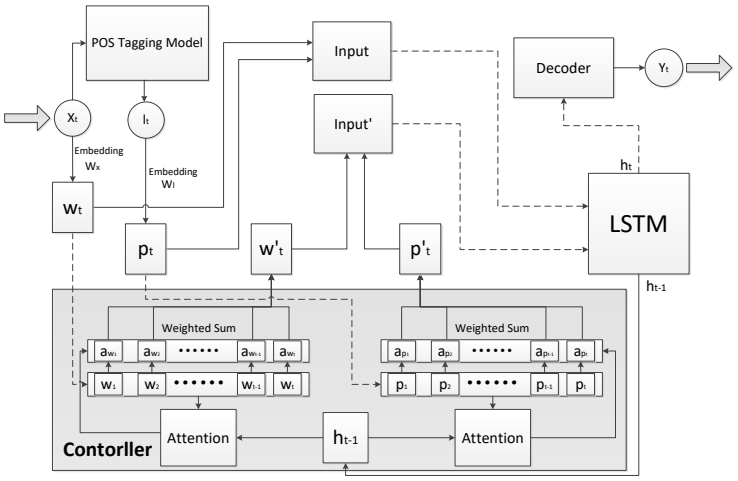

Fig. 2. The network of the deep attentive structured language model with a POS tagging model. The dashed arrows indicate that the connections are applied with dropout operation.

$$m'_t = [w'_t, p'_t] \tag{19}$$

Feeding m_t and m'_t into the LSTM cell instead of w_t and w'_t in equ. 9 to 12, we can compute the hidden and the memory vectors h_t, c_t.

3.3 Integration with LSTMN

LSTMN model [6] stores the entire hidden vectors, and is able to select useful content with the attention layer. In this section, we introduce a simple method to integrate our model with the LSTMN model.

We can score the relation between the current input and the hidden history according to the equ. 1, just replacing x_t by m_t if the POS tagging model is applied, where m_t is the concatenation input of the word and its POS tag. Thus, both $\tilde{h}_{t-1}$ and $\tilde{c}_{t-1}$ can be computed. Using $\tilde{h}_{t-1}, \tilde{c}_{t-1}$ to update the LSTM cell, we can further improve the performances of the language model. The integration model is called DAS-LSTMN in the paper. The results of the experiments comparing these attention based models are shown in the section 4.2.

4 Experiments

4.1 Experimental Setup

Our model is evaluated on the English Penn TreeBank (PTB) [16], Chinese short message (SMS) and Swb-Fisher corpora. The Penn TreeBank corpus is

a famous English dataset, with a vocabulary size of 10K and 4.8% words out of vocabulary (OOV), which is widely used to evaluate the performance of a language model. The training set contains approximately 42K sentences with 887K words. The Chinese SMS corpus is collected from short messages. The corpus has a vocabulary size of about 30K. The training set contains 380K sentences with 1931K words. The Swb-Fisher is an English corpus containing over 26 million words and 2 million sentences. The corpus has a vocabulary size of about 63K. The training set consists of approximately 2.4M sentences with 26.5M words.

For the motivation to improve word-level language model, we only considered the word-level tags such as part of speech (POS). Stanford NLP tools [17] have been utilized to generate the POS tags.

In language modeling, we also use dropout technology [18] for regularizing the networks. In the following experiments, different attention based models are evaluated. We apply different tagging models to the baseline LSTM language model respectively to observe the performances in language modeling. Experiments of our attention based language model combined with the the POS tagging model are also carried out. Finally, experiments in language modeling are conducted respectively on PTB, SMS and Swb-Fisher corpora.

4.2 Experiments on the Attention Mechanisms

As mentioned in the section 3.3, experiments on the attention based model are conducted. In the experiments, we compare the LSTMN[1] [6], the DAS model, and the DAS-LSTMN model without the POS tagging model. The hidden size of all the models are set to 300. We use stochastic gradient descent (SGD) for optimization. Batch size is set to 128 and dropout rate is set to 0.5 at the stage of training.

Table 1. Performances of different attention based models without the POS tagging model

Model	PPL		PPL(Dropout)	
	valid	test	valid	test
LSTM	120.0	114.9	91.3	88.1
LSTMN [6]	114.9	110.4	90.9	87.0
DAS LM	114.7	111.2	90.5	86.6
DAS-LSTMN	**113.6**	**109.1**	**88.8**	**85.3**

The table 1 shows the DAS model outperforms the baseline LSTM and performs as well as the LSTMN. The DAS-LSTMN model gains the best PPL score on PTB corpus, which indicates the integration is effective.

[1] The LSTMN language model is implemented by us. The PPL score in the original paper is 108.

4.3 Experiments on Different POS Tagging Models

We have trained two unidirectional tagging models of different network sizes, using the POS tags generated by Stanford bidirectional tagging tools. The model(a) contains about 3.7M parameters while the model(b) contains about 1.1M parameters. The table 2 shows the tagging performances on PTB corpus.

Table 2. Tagging performances of the unidirectional POS tagging models on PTB corpus

Model	Size	Accuracy(%)
Model(a)	3.7M	96.4
Model(b)	1.1M	92.4

We apply the Stanford tagging tools and these two unidirectional POS tagging models to the baseline LSTM LM respectively to observe the improvements contributed by POS tags. We concatenate the word embedding and the POS tag embedding as the combined input for the baseline LM. The baseline language model is a 1-layer LSTM LM with a hidden size of 300. The table 3 shows the performances of the baseline language model with different POS tagging models.

Table 3. Performances of the baseline LSTM LM with different tagging models on PTB corpus

Model	PPL		PPL(Dropout)	
	valid	test	valid	test
baseline LSTM	120.0	114.9	91.3	88.1
LSTM + Stanford tools	**110.6**	**107.1**	**85.8**	**83.2**
LSTM + tagging model(a)	115.0	111.7	90.9	87.9
LSTM + tagging model(b)	116.2	112.9	91.3	88.9

The table 3 shows the POS tags do help to promote the performances in language modeling. The language model with the Stanford tools, which are based on bidirectional models, achieves the best performances. However, the generated POS tags carry future information, which is not suitable for most language modeling tasks. Tags of this kind may be useful in particular language modeling tasks such as rescoring in speech recognition. The language model with the unidirectional tagging model doesn't utilize future information, but still gains obvious improvement in the experiments. The language model with the tagging model(a) gains the higher PPL score than the model(b), which also achieves the better accuracy score in POS tagging task. In the following experiments, we utilize model(a) as the tagging model.

4.4 Experiments on the Attention Based Models Combined with the POS Tagging Model

POS tags and attention mechanisms prove to be effective in promoting performances in language modeling according to the experiments above. We conduct the experiments of combining these two methods in language modeling.

Table 4. Performances of the attention based models with the POS tagging model on PTB corpus

Model	PPL		PPL(Dropout)	
	valid	test	valid	test
baseline LSTM	120.0	114.9	91.3	88.1
DAS LM	114.7	111.2	90.5	86.6
DAS LM + tagging model	113.3	110.3	89.4	86.4
DAS-LSTMN	113.6	109.1	88.8	85.3
DAS-LSTMN + tagging model	**110.9**	**108.3**	**88.3**	**85.0**

The table 4 shows the attention based language model still achieves minor improvements when combined with our tagging model. The contribution of POS tags to the language model relates highly to the performance of the tagging model. The promotion may be obvious if a better POS tagging model can be applied.

4.5 Experiments in Language Modeling on Different Corpora

Our models are evaluated on PTB, SMS and Swb-Fisher corpora respectively. The compared models include the 4-gram language model with kneser-Ney smoothing (KN4), the baseline LSTM LM, DAS-LSTMN with the POS tagging model and it averagely interpolated with the LSTM LM. For the fair comparison, the hidden size of all the models is set to 300. We use SGD for optimization. Batch size is set to 128 and dropout rate is set to 0.5 at the stage of training. Table 5 shows the performances of different LMs on the three corpora.

The table 5 shows that the DAS-LSTMN model achieves the best PPL scores among all the single models. Averagely interpolated with the conventional LSTM LM, the model achieves consistent and significant improvements on both validation and test sets on all the corpora. The improvements on SMS corpus is not so obvious as the others, for the reason that the sentences of SMS corpus may be too short. The PPL scores are reduced by about 8.4%, 4.8% and 9.4% compared to the conventional LSTM model on the three corpora respectively, which also indicates our model behaves well on various kinds of datasets.

We also take an example from the PTB corpus to observe the relation between the words extracted by the DAS-LSTMN model.

Fig. 3 shows an example sentence from the PTB test set. The arrows indicate high attention scores between the predicted word and its history. We can see most

Table 5. Performances of different LMs on PTB, SMS and Swb-Fisher corpora

Corpus	Model	PPL		PPL(Dropout)	
		valid	test	valid	test
PTB	KN4	151.1	143.6	–	–
	LSTM	120.0	114.9	91.3	88.1
	DAS-LSTMN	110.9	108.3	88.3	85.0
	DAS-LSTMN + LSTM	**97.4**	**94.5**	**84.2**	**80.7**
SMS	KN4	123.1	106.8	–	–
	LSTM	112.5	102.2	102.4	95.3
	DAS-LSTMN	109.4	100.0	101.2	93.0
	DAS-LSTMN + LSTM	**101.2**	**92.3**	**98.2**	**90.8**
Swb-Fisher	KN4	80.9	83.4	–	–
	LSTM	45.8	58.3	**50.2**	63.1
	DAS-LSTMN	49.5	56.5	78.1	65.3
	DAS-LSTMN + LSTM	**43.0**	**52.8**	55.5	**62.4**

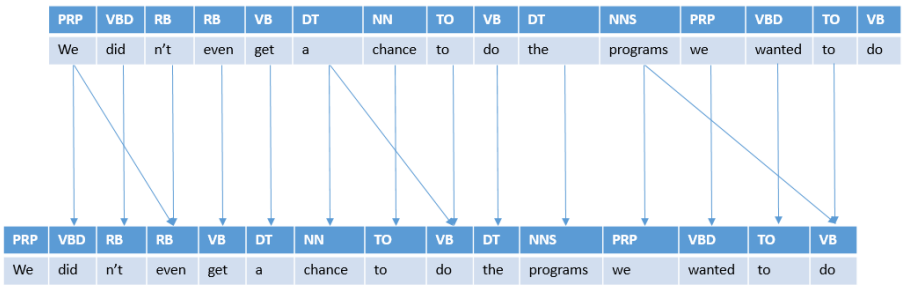

Fig. 3. An example from the PTB test set. The arrows indicate the attention scores higher than 0.2.

words are highly related to the previous word, which agrees with the linguistic intuition that long-term dependencies are relatively rare [6].

5 Conclusion

For the limitation of fixed memory space for conventional RNN based language model, we propose a deep attentive structured language model in this paper. The model is capable of extracting the relation between the hidden state and its input history, applied with the attention module. POS tags are applied as an alternative input of our attention based model. The experiments show that the attention mechanisms do help to achieve improvements in language modeling and POS tags can further promote the language model. The experiments in language modeling indicate that our model outperforms other language models. Interpolated with the LSTM LM, the DAS-LSTMN model achieves significant improvements on the PPL scores. The PPL scores are reduced by 8.4%, 4.8% and 9.4% respectively compared to the baseline LSTM. The results of the experiments on PTB, SMS and Swb-Fisher corpora also indicate our model behaves well on various kinds of datasets.

6 Acknowledge

This work was supported by the Shanghai Sailing Program No. 16YF1405300, the China NSFC projects (No. 61573241 and No. 61603252) and the Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China. Experiments have been carried out on the PI supercomputer at Shanghai Jiao Tong University.

7 The References Section

References

1. Bengio, Yoshua, et al.: A neural probabilistic language model. Journal of machine learning research 3.Feb, 1137-1155 (2003)
2. Mikolov, Tomas, et al.: Recurrent neural network based language model. Interspeech. Vol. 2. (2010)
3. Mikolov, Tom, et al.: Extensions of recurrent neural network language model. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on. IEEE (2011)
4. Hochreiter, Sepp, and Jrgen Schmidhuber.: Long short-term memory. Neural computation 9.8 1735-1780 (1997)
5. Sundermeyer, Martin, Ralf Schlter, and Hermann Ney.: LSTM Neural Networks for Language Modeling. Interspeech (2012)
6. Cheng Jianpeng, Li Dong, and Mirella Lapata.: Long short-term memory-networks for machine reading. arXiv preprint arXiv:1601.06733 (2016)

7. Chelba, Ciprian, and Frederick Jelinek.: Structured language modeling for speech recognition. arXiv preprint cs/0001023 (2000)
8. Hochreiter, Sepp, et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies (2001)
9. Graves, Alex, Greg Wayne, and Ivo Danihelka.: Neural turing machines. arXiv preprint arXiv:1410.5401 (2014)
10. Sukhbaatar, Sainbayar, Jason Weston, and Rob Fergus.: End-to-end memory networks. Advances in neural information processing systems (2015)
11. Cho K, Courville A, Bengio Y.: Describing Multimedia Content Using Attention-Based Encoder-Decoder Networks[J]. IEEE Transactions on Multimedia, 17(11):1875-1886 (2015)
12. Feng S, Liu S, Li M, et al.: Implicit distortion and fertility models for attention-based encoder-decoder NMT model[J]. arXiv preprint arXiv:1601.03317 (2016)
13. Bahdanau D, Cho K, Bengio Y: Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473 (2014)
14. Part of speech `https://en.wikipedia.org/wiki/Part_of_speech`
15. Wang, Peilu, et al.: A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. arXiv preprint arXiv:1511.00215 (2015)
16. Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini.: Building a large annotated corpus of English: The Penn Treebank. Computational linguistics 19.2:313-330 (1993)
17. The Stanford Natural Language Processing Group `https://nlp.stanford.edu/software/stanford-dependencies.shtml`
18. Zaremba W, Sutskever I, Vinyals O.: Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329 (2014)