AAAI-19

Long Short-Term Memory with Dynamic Skip Connections

Tao Gui, *Qi Zhang, Lujun Zhao, Yaosong Lin, Minlong Peng, Jingjing Gong, Xuanjing Huang*

Fudan University

復旦大學

FUDAN UNIVERSITY

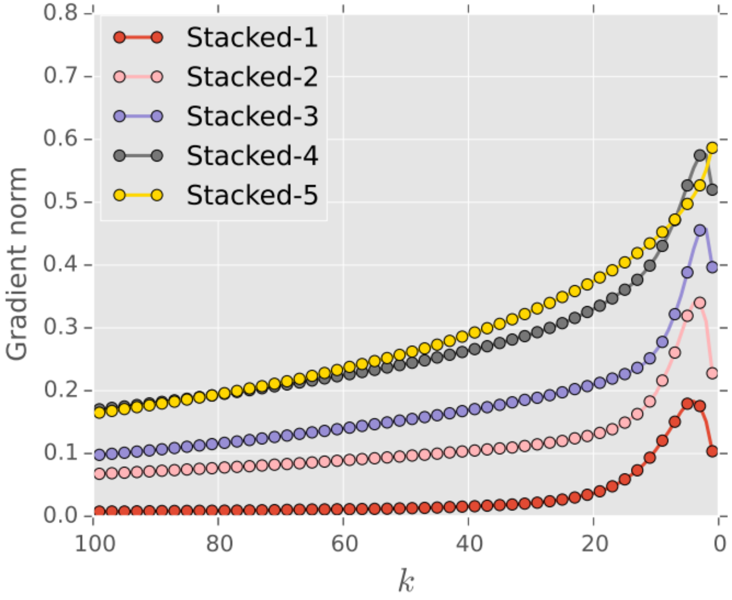
2019.1.12

Outline

1. Introduction and Motivation
2. Our Approach
3. Experiments
4. Conclusion

Introduction and Motivation

Background



Long-term effect of the cell states on the loss function. The average value of $\left\| \frac{\partial L_t}{\partial c_{t-k}} \right\|$,

Mujika, Asier, Florian Meier, and Angelika Steger. "Fast-slow recurrent neural networks." *Advances in Neural Information Processing Systems*. 2017.

Introduction and Motivation

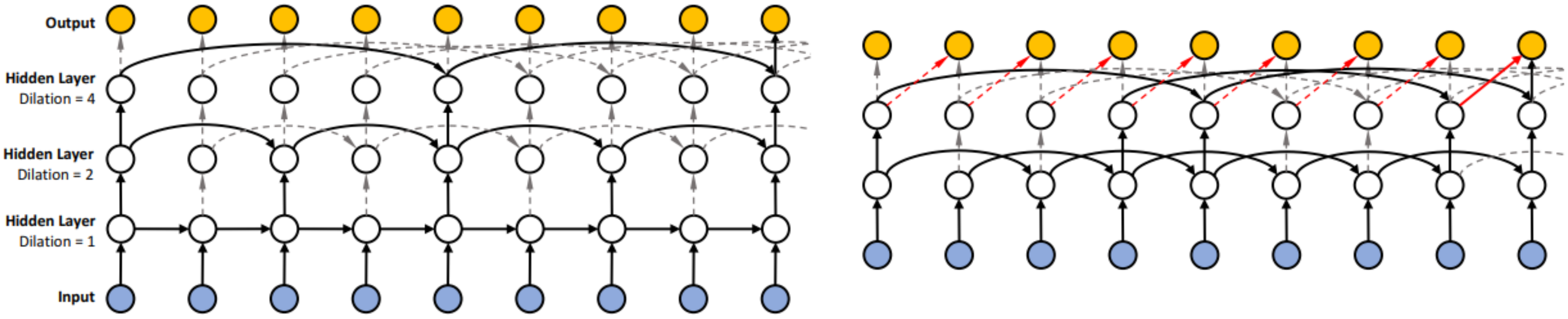


Figure 2: (left) An example of a three-layer DILATEDRNN with dilation 1, 2, and 4. (right) An example of a two-layer DILATEDRNN, with dilation 2 in the first layer. In such a case, extra embedding connections are required (red arrows) to compensate missing data dependencies.

Chang, Shiyu, et al. "Dilated recurrent neural networks." *Advances in Neural Information Processing Systems*. 2017.

Introduction and Motivation

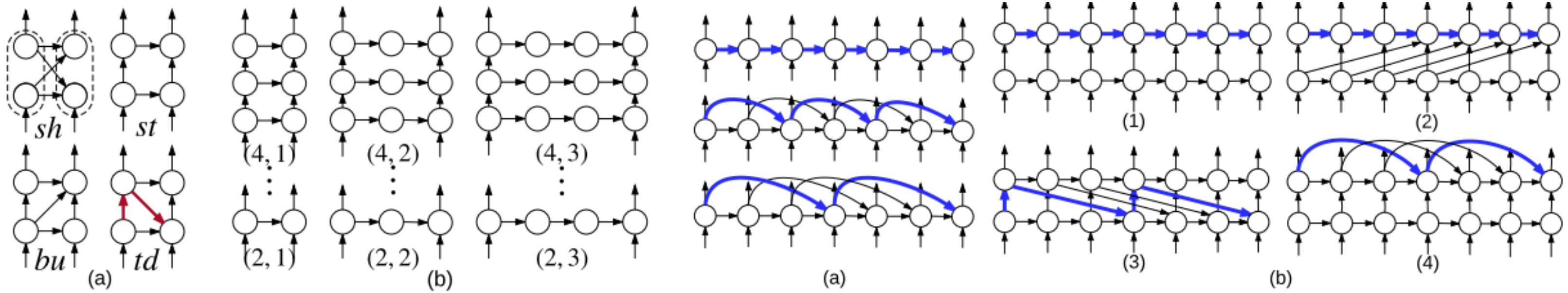
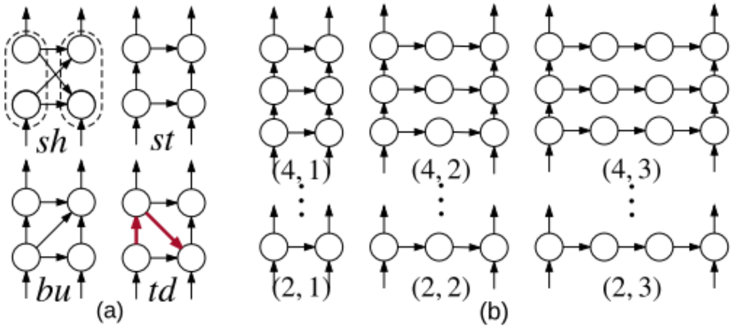
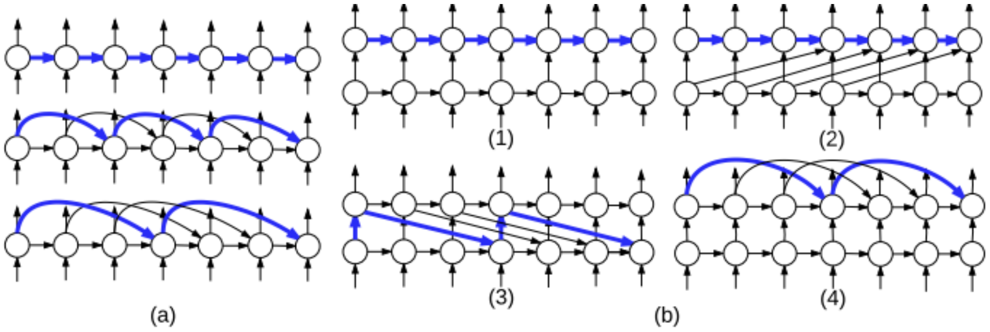


Figure 2: **Left**: (a) The architectures for sh, st, bu and td, with their (d_r, d_f) equal to $(1, 2)$, $(1, 3)$, $(1, 3)$ and $(2, 3)$, respectively. The longest path in td are colored in red. (b) The 9 architectures denoted by their (d_f, d_r) with $d_r = 1, 2, 3$ and $d_f = 2, 3, 4$. In both (a) and (b), we only plot hidden states at two adjacent time steps and the connections between them (the period number is 1). **Right**: (a) Various architectures that we consider in Section 4.4. From top to bottom are baseline $s = 1$, and $s = 2$, $s = 3$. (b) Proposed architectures that we consider in Section 4.5 where we take $k = 3$ as an example. The shortest paths in (a) and (b) that correspond to the recurrent skip coefficients are colored in blue.

Zhang, Saizheng, et al. "Architectural complexity measures of recurrent neural networks." *Advances in Neural Information Processing Systems*. 2016.

Introduction and Motivation



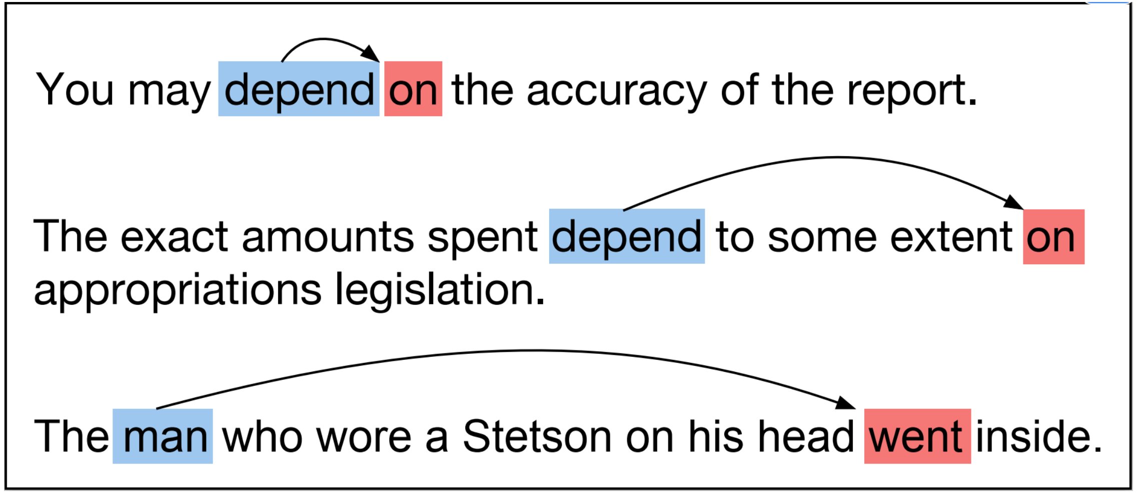


They found empirical evidences that increasing **feedforward depth might not** help on long term dependency tasks, while increasing the **recurrent skip coefficient can largely improve** performance on long term dependency tasks.

Zhang, Saizheng, et al. "Architectural complexity measures of recurrent neural networks." *Advances in Neural Information Processing Systems*. 2016.

Introduction and Motivation

Challenges



You may depend on the accuracy of the report.

The exact amounts spent depend to some extent on appropriations legislation.

The man who wore a Stetson on his head went inside.

Figure 1: Examples of dependencies with variable length in the language.

Outline

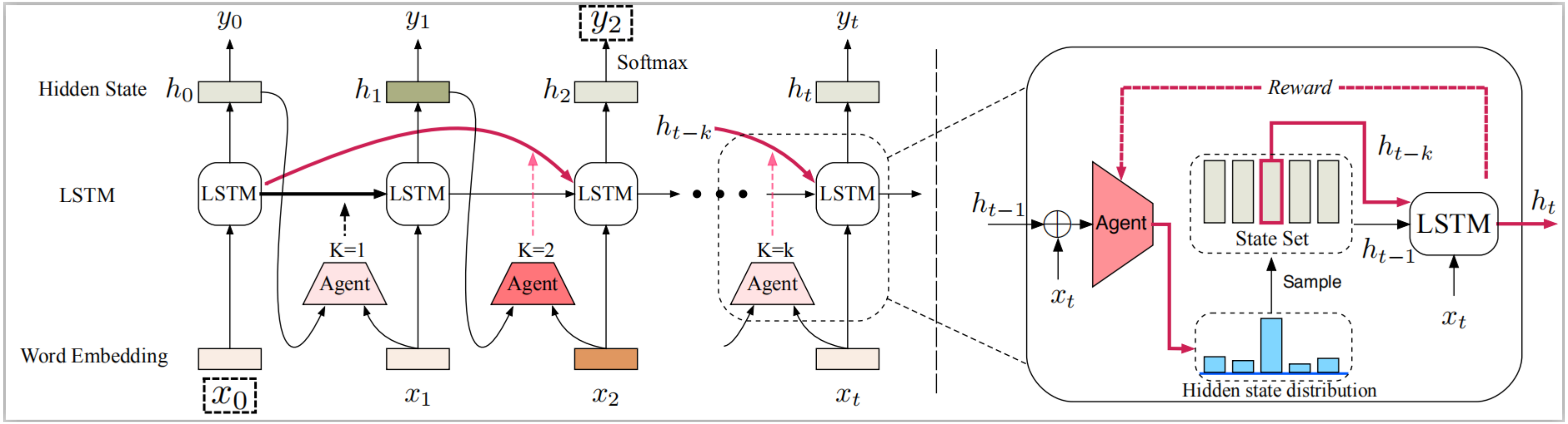
1 Introduction and Motivation

2 Our Approach

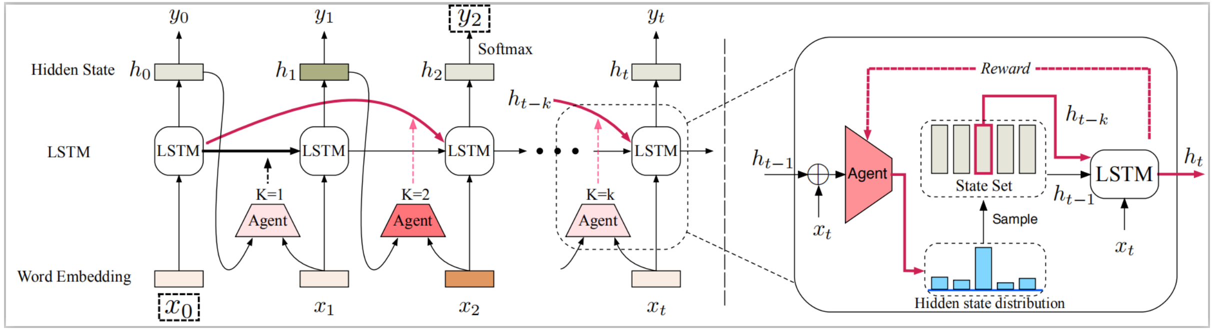
3 Experiments

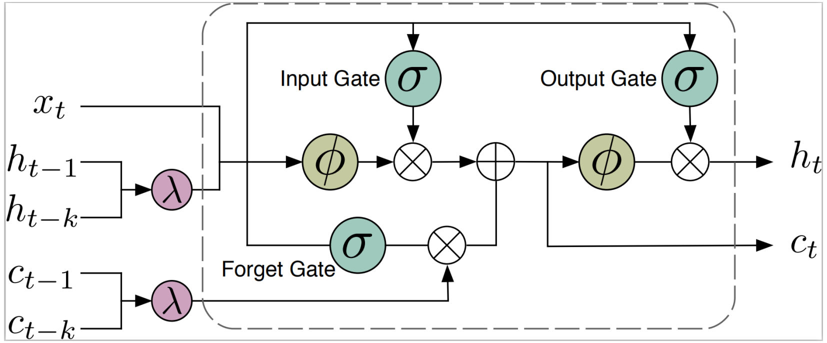
4 Conclusion

System Framework



System Framework





$$\widetilde{\mathbf{h}}_{t-1} = \lambda \mathbf{h}_{t-k} + (1-\lambda)\mathbf{h}_{t-1}$$

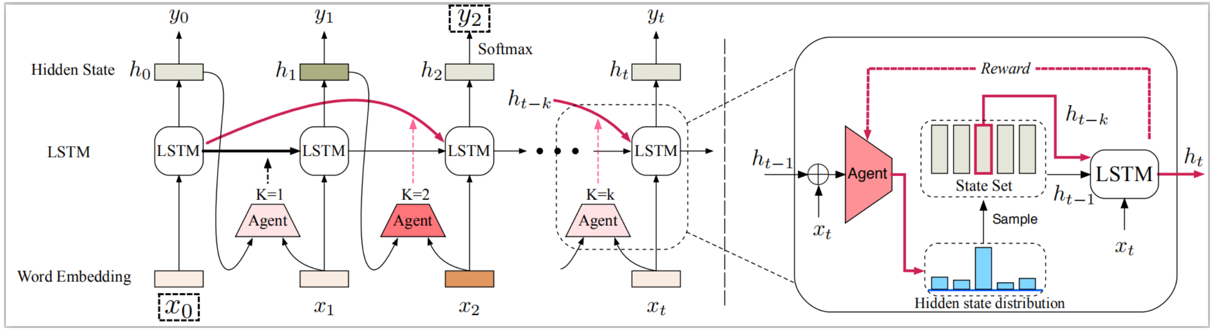
$$\widetilde{\mathbf{c}}_{t-1} = \lambda \mathbf{c}_{t-k} + (1-\lambda)\mathbf{c}_{t-1}$$

$$\begin{bmatrix} \mathbf{g}_t \\ \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \end{bmatrix} = \left(\begin{bmatrix} \mathbf{W}_x^g, \mathbf{W}_h^g \\ \mathbf{W}_x^i, \mathbf{W}_h^i \\ \mathbf{W}_x^f, \mathbf{W}_h^f \\ \mathbf{W}_x^o, \mathbf{W}_h^o \end{bmatrix} \bullet \begin{bmatrix} \mathbf{x}_t \\ \widetilde{\mathbf{h}}_{t-1} \end{bmatrix} + \begin{bmatrix} \mathbf{b}^g \\ \mathbf{b}^i \\ \mathbf{b}^f \\ \mathbf{b}^o \end{bmatrix} \right)$$

$$\mathbf{c}_t = \phi(\mathbf{g}_t) \odot \sigma(\mathbf{i}_t) + \widetilde{\mathbf{c}}_{t-1} \odot \sigma(\mathbf{f}_t)$$

$$\mathbf{h}_t = \sigma(\mathbf{o}_t) \odot \phi(\mathbf{c}_t)\,,$$

System Framework



$$J_1(\theta_l) = -[y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)],$$

$$J_2(\theta_a) = \mathbb{E}_{\pi(a_{1:T})}[R] + H(\pi(a_{1:T})),$$

$$\nabla_{\theta_a} J_2(\theta_a) = \mathbb{E}_{\pi(a_{1:T})}[\sum_{t=1}^{T} \nabla_{\theta_a} \log Pr(a_t|s_t;\theta_a)*$$

$$(R - \sum_{t=1}^{T} \log Pr(a_t|s_t;\theta_a) - 1)].$$

Outline

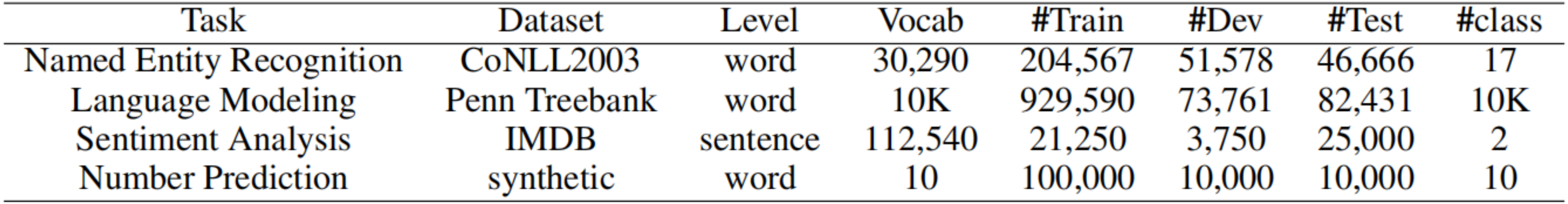
1 Introduction and Motivation

2 Our Approach

3 Experiments

4 Conclusion

Experiments



Task	Dataset	Level	Vocab	#Train	#Dev	#Test	#class
Named Entity Recognition	CoNLL2003	word	30,290	204,567	51,578	46,666	17
Language Modeling	Penn Treebank	word	10K	929,590	73,761	82,431	10K
Sentiment Analysis	IMDB	sentence	112,540	21,250	3,750	25,000	2
Number Prediction	synthetic	word	10	100,000	10,000	10,000	10

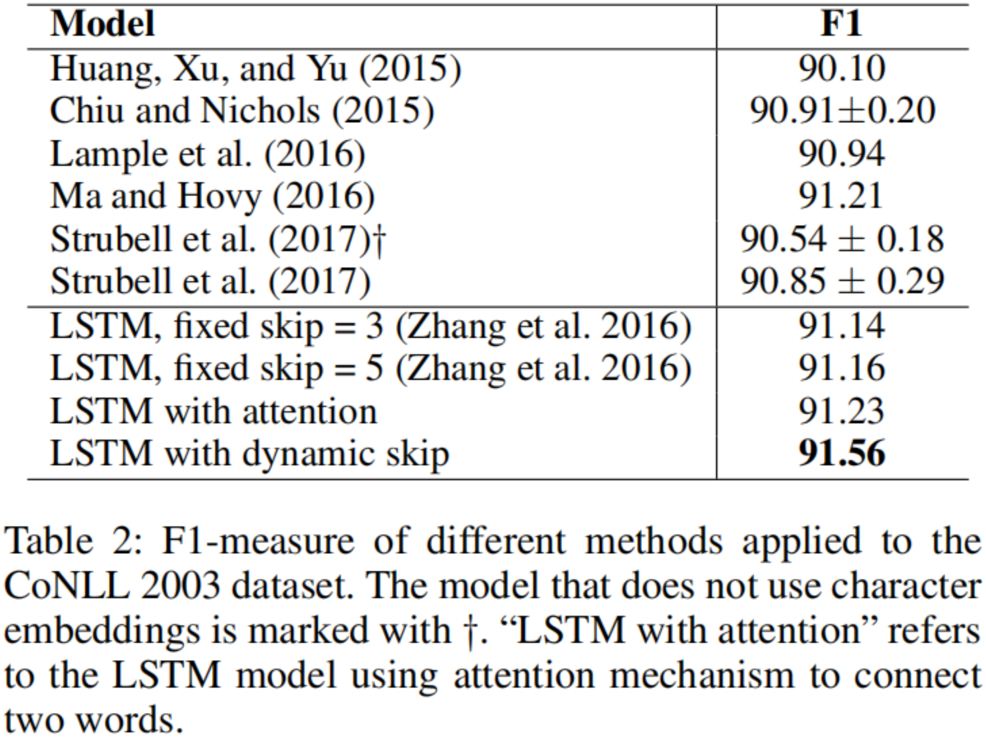
Table 1: Statistics of the CoNLL2003, Penn Treebank, IMDB, and synthetic datasets.

Tjong Kim Sang, Erik F., and Fien De Meulder. "Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition." *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*. Association for Computational Linguistics, 2003.

Marcus, Mitchell P., Mary Ann Marcinkiewicz, and Beatrice Santorini. "Building a large annotated corpus of English: The Penn Treebank." Computational linguistics 19.2 (1993): 313-330.

Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1. Association for Computational Linguistics, 2011.

Experiments



Model	F1
Huang, Xu, and Yu (2015)	90.10
Chiu and Nichols (2015)	90.91±0.20
Lample et al. (2016)	90.94
Ma and Hovy (2016)	91.21
Strubell et al. (2017)†	90.54 ± 0.18
Strubell et al. (2017)	90.85 ± 0.29
LSTM, fixed skip = 3 (Zhang et al. 2016)	91.14
LSTM, fixed skip = 5 (Zhang et al. 2016)	91.16
LSTM with attention	91.23
LSTM with dynamic skip	**91.56**

Table 2: F1-measure of different methods applied to the CoNLL 2003 dataset. The model that does not use character embeddings is marked with †. “LSTM with attention” refers to the LSTM model using attention mechanism to connect two words.

Experiments

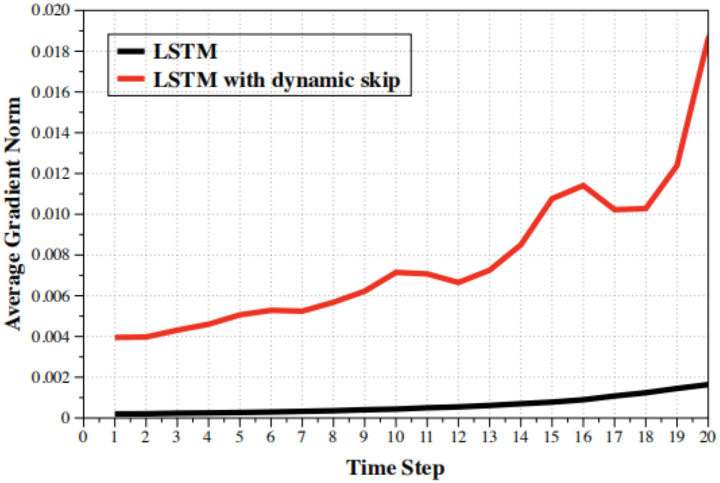
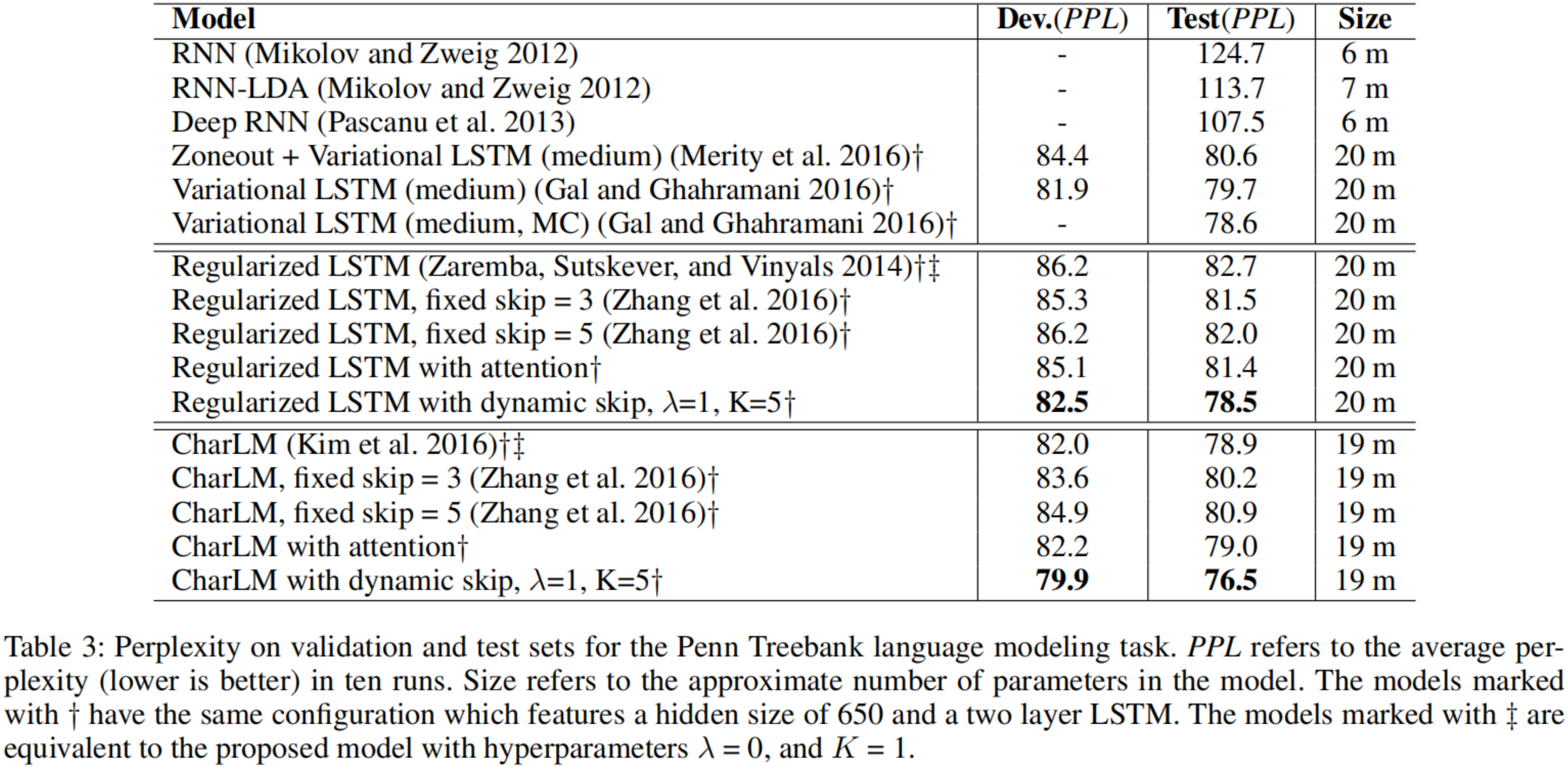


Figure 4: Normalized long-term gradient values $\left\|\frac{\partial L_T}{\partial h_t}\right\|$ tested on CoNLL 2003 dataset. At the initial time steps, the proposed model still preserves effective gradients, which is hundreds of times larger than those in the standard LSTM, indicating that the proposed model have stronger ability to capture long-term dependency.

Experiments



Model	Dev.(*PPL*)	Test(*PPL*)	Size
RNN (Mikolov and Zweig 2012)	-	124.7	6 m
RNN-LDA (Mikolov and Zweig 2012)	-	113.7	7 m
Deep RNN (Pascanu et al. 2013)	-	107.5	6 m
Zoneout + Variational LSTM (medium) (Merity et al. 2016)†	84.4	80.6	20 m
Variational LSTM (medium) (Gal and Ghahramani 2016)†	81.9	79.7	20 m
Variational LSTM (medium, MC) (Gal and Ghahramani 2016)†	-	78.6	20 m
Regularized LSTM (Zaremba, Sutskever, and Vinyals 2014)†‡	86.2	82.7	20 m
Regularized LSTM, fixed skip = 3 (Zhang et al. 2016)†	85.3	81.5	20 m
Regularized LSTM, fixed skip = 5 (Zhang et al. 2016)†	86.2	82.0	20 m
Regularized LSTM with attention†	85.1	81.4	20 m
Regularized LSTM with dynamic skip, λ=1, K=5†	**82.5**	**78.5**	20 m
CharLM (Kim et al. 2016)†‡	82.0	78.9	19 m
CharLM, fixed skip = 3 (Zhang et al. 2016)†	83.6	80.2	19 m
CharLM, fixed skip = 5 (Zhang et al. 2016)†	84.9	80.9	19 m
CharLM with attention†	82.2	79.0	19 m
CharLM with dynamic skip, λ=1, K=5†	**79.9**	**76.5**	19 m

Table 3: Perplexity on validation and test sets for the Penn Treebank language modeling task. *PPL* refers to the average perplexity (lower is better) in ten runs. Size refers to the approximate number of parameters in the model. The models marked with † have the same configuration which features a hidden size of 650 and a two layer LSTM. The models marked with ‡ are equivalent to the proposed model with hyperparameters $\lambda = 0$, and $K = 1$.

Experiments

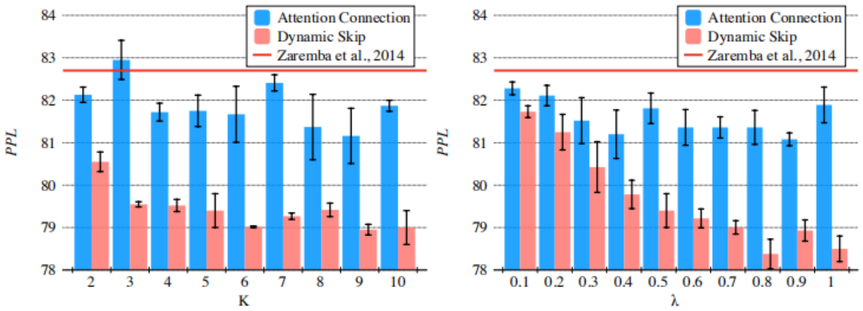
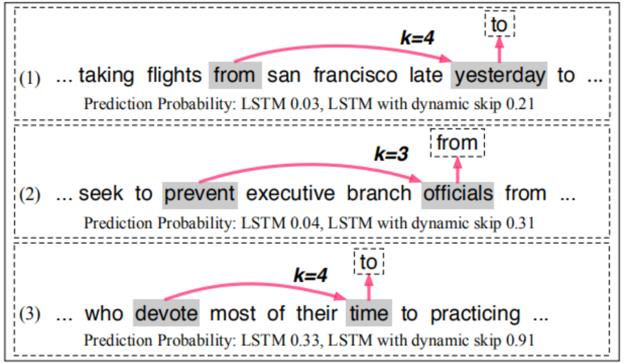


Figure 5: Test set perplexities (lower is better) on Penn Treebank language model corpus with standard deviation for K from 2 to 10 with $\lambda = 0.5$ (left), and λ from 0.1 to 1.0 with $K = 5$ (right).



(1) ... taking flights from san francisco late yesterday to ...

k=4

to

Prediction Probability: LSTM 0.03, LSTM with dynamic skip 0.21

(2) ... seek to prevent executive branch officials from ...

k=3

from

Prediction Probability: LSTM 0.04, LSTM with dynamic skip 0.31

(3) ... who devote most of their time to practicing ...

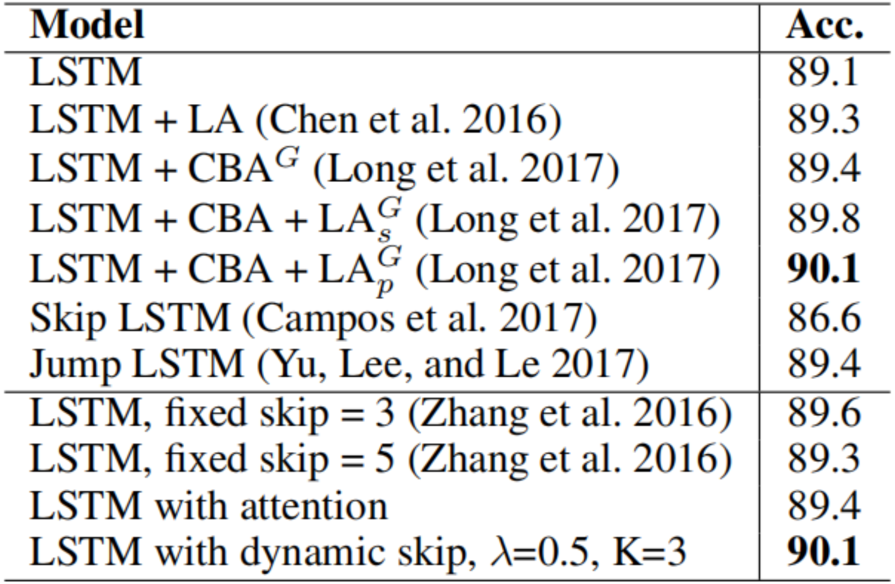
k=4

to

Prediction Probability: LSTM 0.33, LSTM with dynamic skip 0.91

Figure 6: Examples of the proposed model applied to language modeling.

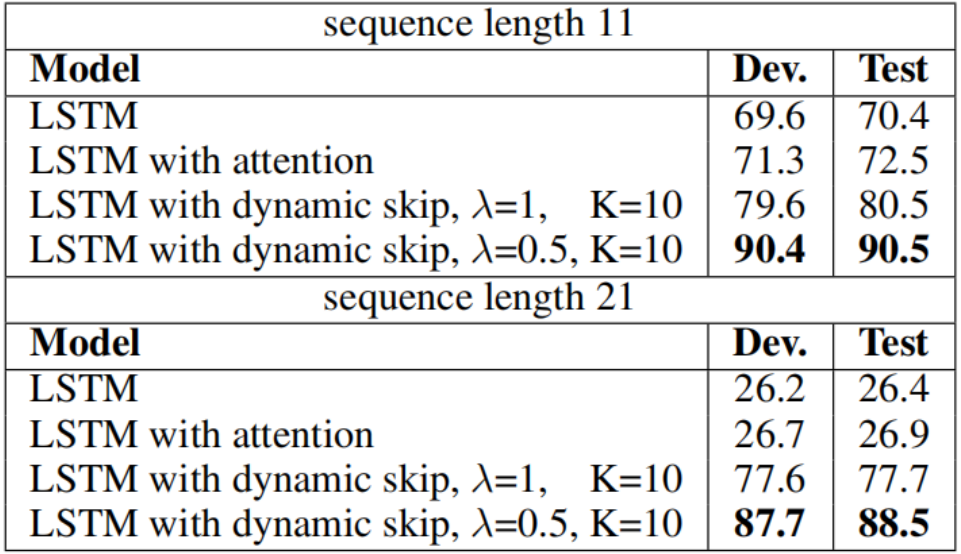
Experiments



Model	**Acc.**
LSTM	89.1
LSTM + LA (Chen et al. 2016)	89.3
LSTM + CBA^G (Long et al. 2017)	89.4
LSTM + CBA + LA$_s^G$ (Long et al. 2017)	89.8
LSTM + CBA + LA$_p^G$ (Long et al. 2017)	**90.1**
Skip LSTM (Campos et al. 2017)	86.6
Jump LSTM (Yu, Lee, and Le 2017)	89.4
LSTM, fixed skip = 3 (Zhang et al. 2016)	89.6
LSTM, fixed skip = 5 (Zhang et al. 2016)	89.3
LSTM with attention	89.4
LSTM with dynamic skip, λ=0.5, K=3	**90.1**

Table 4: Accuracy on the IMDB test set.

Experiments



sequence length 11		
Model	**Dev.**	**Test**
LSTM	69.6	70.4
LSTM with attention	71.3	72.5
LSTM with dynamic skip, λ=1, K=10	79.6	80.5
LSTM with dynamic skip, λ=0.5, K=10	**90.4**	**90.5**
sequence length 21		
Model	**Dev.**	**Test**
LSTM	26.2	26.4
LSTM with attention	26.7	26.9
LSTM with dynamic skip, λ=1, K=10	77.6	77.7
LSTM with dynamic skip, λ=0.5, K=10	**87.7**	**88.5**

Table 5: Accuracies of different methods on number prediction dataset.

Outline

1 Introduction and Motivation

2 Our Approach

3 Experiments

4 Conclusion

Conclusion

1 We study the sequence modeling problem incorporating dynamic skip connections, which can effectively tackle the long-term dependency problems.

2 We propose a novel reinforcement learning-based LSTM model to achieve the task, and the proposed model can learn to choose one optimal set of hidden and cell states from the past few states.

3 Several experiment results are given to demonstrate the effectiveness of the proposed method from different aspects.

THANK YOU

復旦大學

FUDAN UNIVERSITY