FUTURE VECTOR ENHANCED LSTM LANGUAGE MODEL FOR LVCSR

Qi Liu, Yanmin Qian, 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
Emails: {liuq901, yanminqian, kai.yu}@sjtu.edu.cn

ABSTRACT

Language models (LM) play an important role in large vocabulary continuous speech recognition (LVCSR). However, traditional language models only predict next single word with given history, while the consecutive predictions on a sequence of words are usually demanded and useful in LVCSR. The mismatch between the single word prediction modeling in trained and the long term sequence prediction in read demands may lead to the performance degradation. In this paper, a novel enhanced long short-term memory (LSTM) LM using the future vector is proposed. In addition to the given history, the rest of the sequence will be also embedded by future vectors. This future vector can be incorporated with the LSTM LM, so it has the ability to model much longer term sequence level information. Experiments show that, the proposed new LSTM LM gets a better result on BLEU scores for long term sequence prediction. For the speech recognition rescoring, although the proposed LSTM LM obtains very slight gains, the new model seems obtain the great complementary with the conventional LSTM LM. Rescoring using both the new and conventional LSTM LMs can achieve a very large improvement on the word error rate.

Index Terms: speech recognition, language model, recurrent neural network, n-best rescoring

1. INTRODUCTION

Language model plays an important role in LVCSR. N-gram [1, 2] has been widely used in the LVCSR system for a long time. However, n-gram only uses limited histories which is hard to deal with long context sequences. RNN and LSTM language models [3, 4] which can store the whole history of the sequence have been proposed to deal with this problem and obtained great success in many fields [5, 6].

However, many sequence level tasks including machine translation [7], speech recognition [8] and handwriting recognition [9] need long term sequence prediction, while the traditional RNN language model only predicts single word one by one. According to [10], there is a gap between the common used word level metric perplexity (PPL) for language model evaluation and the true sequence level metric such like BLEU score in machine translation [11] and word error rate (WER) in speech recognition [12].

Several researches have been done to deal with this problem. [13, 14, 15] researched on training bidirectional LSTM language model, which can retrieve the the information not only from the past context but also the future context. [10] combined reinforcement learning and deep learning together, directly trained the neural network with the estimated BLEU score. [16, 17] applied sequence to sequence training method on language model.

In this paper, an novel enhanced LSTM language model has been proposed. Enhanced LSTM language model predicts not only a single word, but also the whole future of the input sequence. It is believed that enhanced LSTM language model can perform well with more sequence level information.

Enhanced LSTM language model trains a reversed LSTM language model. And the activation values of the last hidden layer of this reversed LSTM are used as bottleneck features [18] which can embed the future of the sequence. These bottleneck features are called future vectors of the sequence.

These future vectors which contain sequence level information will be used to train the enhanced LSTM language model. The model will be trained by not only to predict the next word but also the future vector. The predicted future vector will also be the input feature to predict the next word.

The experiments show that the enhanced LSTM language model performs well on the sequence prediction task. It is also observed that in n-best rescoring task, the WER can get a very large improvement by the combination on the normal and enhanced LSTM language model.

The rest of the paper is organized as follows, section 2 is the background. Section 3 indicates the methodology of enhanced LSTM language model and section 4 shows the experimental setup and results. Finally, conclusion will be given in section 5 and discussion can be found in section 6.

978-1-5090-4788-8/17/$31.00 ©2017 IEEE

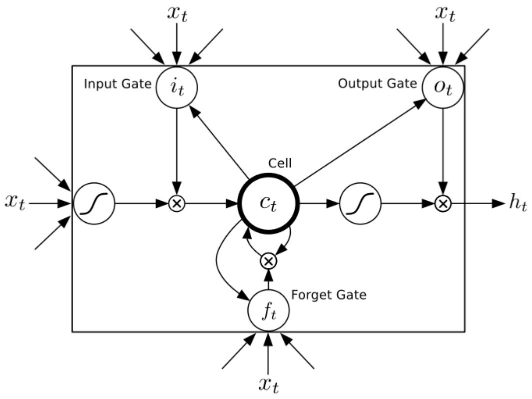

Fig. 1. One LSTM memory cell [25]. There are three gates (input gate, output gate and forget gate) in each cell to control the data flow. In practice, h_{t-1} will also be the input to the cell together with x_t.

2. BACKGROUND

2.1. Long Short-Term Memory

RNN [19] is the neural network with cycles in its structure, which is effective in dealing with sequential data. Suppose there is a sequence of data $x_1, x_2, \ldots, x_T$ as the input and let $h_1, h_2, \ldots, h_T$ be the output of one RNN, the most commonly used RNN formula looks like

$$h_t = f(W_x x_t + W_h h_{t-1} + b).$$

where W_x and W_h are weight matrix parameters, b is the bias and f is the activation.

Due to gradient vanishing and explosion problems [20, 21], LSTM [3], which is a unit structured RNN, has been used to replace the traditional RNN. LSTM-RNN shows better performance [22, 23, 24], and the LSTM formula is shown below:

$$\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f) \\
m_t &= \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\
c_t &= f_t \cdot c_{t-1} + i_t \cdot m_t \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o) \\
h_t &= o_t \cdot \tanh(c_t).
\end{aligned}$$

where W_{**} are the weight matrix parameters, b_* are the bias and σ is the sigmoid function. The detail of its structure can be found in Figure 1.

2.2. LSTM Language Model

LSTM language model uses the current word as the input and the next word as the output. In detail, suppose $x_1, x_2, \ldots, x_T$ is the input sequence, x_i is the i-th word, and the vocabulary size is n. The input layer of the LSTM is a word embedding layer with size n, and the output layer of the LSTM is a softmax layer with size n. The detail formula is shown below:

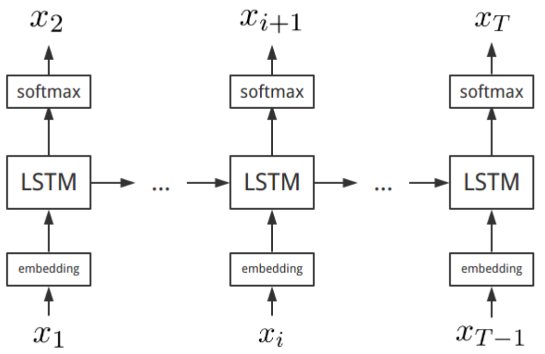

Fig. 2. The structure of LSTM language model. Here $x_1, x_2, \ldots, x_T$ is the input sequence.

$$\begin{aligned}
\bar{x_i} &= f(x_i) \\
h_i &= \text{LSTM}(\bar{x_i}, h_{i-1}) \\
p_i &= \text{softmax}(Wh_i + b) \\
x_{i+1} &= \arg\max p_i,
\end{aligned}$$

where f represents the word embedding and W, b are the network parameters. Figure 2 shows the structure of LSTM language model. At the i-th time step, x_i is the input to the LSTM, and the output value $p_i = (p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(n)})$ is considered to be the probability of observe each word at time step $i + 1$, i.e.

$$p(x_{i+1}|x_1, x_2, \ldots, x_i) = p_i^{(x_{i+1})}.$$

To train the LSTM language model, the cross entropy (CE) of output distribution p_i and the ground truth distribution

$$g_i = (0, \ldots, 0, 1, 0, \ldots, 0|1 \text{ at position } x_{i+1})$$

will be used as the criterion to train the network, i.e. the loss function is

$$\mathcal{L} = \text{CE}(g_i, p_i) = -\sum_{j=1}^{n} g_i^{(j)} \log p_i^{(j)}.$$

3. METHODOLOGY

3.1. Future Vector Extraction

Traditional LSTM language models only predict a single word for the given history, which may lose information about the whole future. In contrast the rest of the sequence will be

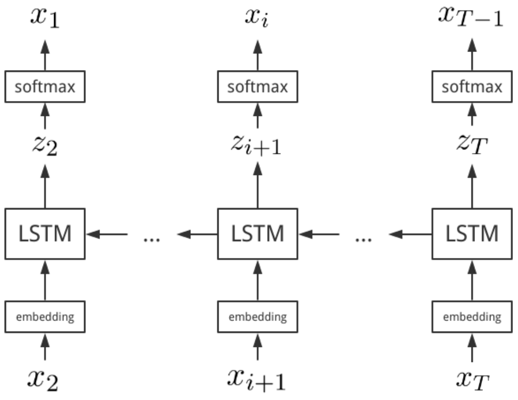

Fig. 3. The structure of future vector extractor. Here $x_1, x_2, \ldots, x_T$ is the input sequence and $z_2, z_3, \ldots, z_T$ are the extracted future vectors.

embedded into a sequence vector in the new proposed enhanced LSTM language model. This sequence vector, which is called future vector in this paper, contains the information about all the sequence future.

There are several ways [26, 27, 28] to extract future vectors. What is needed here is that for a given input sequence, each suffix needs be embedded and the relationship among them must be kept. Therefore the method similar to [29] has been chosen. A normal LSTM language model with reversed input sequence order has been trained, which means this LSTM language model predicts the previous word with the given future. The future vector is extracted from the activation values of the last hidden layer in this reversed LSTM language model. Figure 3 shows the detailed structure and the formula is shown below.

$$\begin{aligned}
\bar{x_i} &= f(x_i) \\
z_i &= \text{LSTM}(\bar{x_i}, z_{i+1}) \\
p_i &= \text{softmax}(Wz_i + b) \\
x_{i-1} &= \arg\max p_i,
\end{aligned}$$

where f is the word embedding and W, b are model parameters. $z_2, z_3, \ldots, z_T$ are the extracted future vectors.

3.2. Enhanced LSTM Language Model

Future vectors cannot be directly used to train a language model. For a input sequence $x_1, x_2, \ldots, x_T$ and its future vectors $z_1, z_2, \ldots, z_T$, only history $x_1, x_2, \ldots, x_i$ are known while the language model is trying to predict word x_{i+1}. However, the future vector z_{i+1} is a function of unknown future $x_{i+1}, x_{i+2}, \ldots, x_T$ which is impossible to be generated.

One additional LSTM network has been trained to solve this problem. This network is similar to normal LSTM language model but predicts the future vector rather than the next word. The detailed formula is

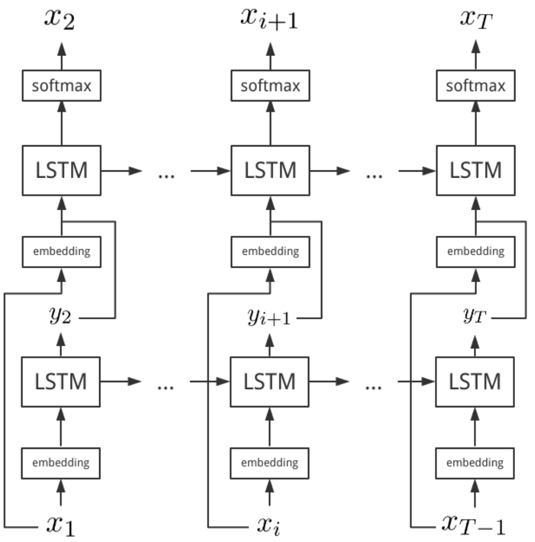

Fig. 4. The structure of enhanced LSTM language model. Here $x_1, x_2, \ldots, x_T$ is the input sequence and $y_2, y_3, \ldots, y_T$ are the predicted future vectors. In practice, the two LSTM networks are trained separately.

$$\begin{aligned}
\bar{x_i} &= f(x_i) \\
h_i &= \text{LSTM}(\bar{x_i}, h_{i-1}) \\
y_{i+1} &= Wh_i + b
\end{aligned}$$

where f is word embedding and W, b are network parameters. The criterion to train this network is the mean squared error (MSE) between the future vector prediction y_i and the truly extracted future vector z_i described in section 3.1, i.e. the error function is

$$\mathcal{L} = \text{MSE}(y_i, z_i) = \frac{1}{m}\sum_{j=1}^{m}(y_i^{(j)} - z_i^{(j)})^2,$$

where m is the dimension of future vector.

y_i is a function of $x_1, x_2, \ldots, x_{i-1}$ which means it can be directly used to train a language model. In enhanced LSTM language model, y_{i+1} will be combined together with x_i as the new input of the LSTM language model, i.e.

$$\begin{aligned}
\bar{x_i} &= f(x_i) \\
h_i &= \text{LSTM}(\bar{x_i}, y_{i+1}, h_{i-1}) \\
p_i &= \text{softmax}(Wh_i + b) \\
x_{i+1} &= \arg\max p_i,
\end{aligned}$$

where f indicates the word embedding and W, b are network parameters. The criterion is CE which is the same as normal LSTM language model in section 2.2. The details structure is illustrated in figure 4.

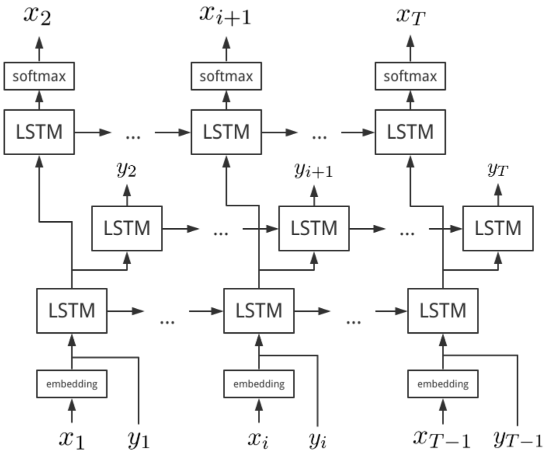

Fig. 5. The structure of multi-task enhanced LSTM language model. Here $x_1, x_2, \ldots, x_T$ is the input sequence and $y_2, y_3, \ldots, y_T$ are the predicted future vectors. y_1 is a zero vector. In practice, the three LSTM networks are trained together.

Enhanced LSTM language model has more input, the future vector y_i, to predict the next word compared with the normal LSTM language model. This results an enhanced LSTM language model which has the power ability to modeling future sequence level information.

3.3. Multi-task Enhanced LSTM

Enhanced LSTM language model has two networks, one is future vector prediction LSTM and the other one is language model LSTM. It is observed that these two networks can be trained together. Multi-task training [30, 31, 32] is a suitable method for joint training.

The prediction of next word and corresponding future vector can be optimized at the same time in the multi-task enhanced LSTM language model. The predicted future vector will also be the input like the non multi-task version. The detailed formula is here,

$$\begin{aligned}
\bar{x_i} &= f(x_i) \\
h_i &= \text{LSTM}(\bar{x_i}, y_i, h_{i-1}) \\
u_i &= \text{LSTM}(h_i, u_{i-1}) \\
y_{i+1} &= W_u u_i + b_u \\
v_i &= \text{LSTM}(h_i, v_{i-1}) \\
p_i &= \text{softmax}(W_v v_i + b_v) \\
x_{i+1} &= \arg\max p_i,
\end{aligned}$$

where f is the word embedding and W_*, b_* are network parameters. The two criteria to train this multi-task network is MSE for future vector prediction and CE for word prediction which also have been used for non multi-task version in section 3.2, i.e. the loss function is

$$\mathcal{L} = \text{CE}(g_i, p_i) + \lambda \text{MSE}(y_{i+1}, z_{i+1}),$$

$\lambda = 1.0$ in this implementation. The structure is Figure 5.

Multi-task enhanced LSTM language model can get not only explicit sequence level information from the input but also the implicit sequence level information from the future vector prediction.

Model	Input	Output
LSTM	x_i	x_{i+1}
FV	x_i x_i, y_{i+1}	y_{i+1} x_{i+1}
MT-FV	x_i, y_i	x_{i+1}, y_{i+1}

Table 1. Brief comparison among three LSTM language model structures. FV indicates the future vector enhanced LSTM, and MT-FV indicates the future vector enhanced LSTM with multi-task training. x_* indicates the original input sequence and y_* is the predicted future vector.

In table 1, a briefly comparison of structures among normal LSTM, enhanced LSTM and multi-task enhanced LSTM language model has been shown.

4. EXPERIMENTS

4.1. Experimental Setup

The experiments are designed to evaluate the performance of the proposed enhanced LSTM language model. The experiments uses two corpora including PTB English corpus and short messages Chinese corpus. PTB corpus contains 49199 utterances and Chinese short messages corpus has 403218 utterances. The vocabulary size is 10000 and 40697 respectively. The experiments used almost the same structure in all the systems. All the LSTM block in Figure 2, 3 and 4 is a stacked three hidden layers LSTM. In Figure 5, the multi-task network has two hidden LSTM layers in shared part and one hidden LSTM layer in separate part. All the LSTM hidden layers contains 300 cells.

Both sequence prediction and speech recognition n-best rescoring will be evaluated, and the BLEU score and WER are used respectively.

4.2. Experimental Results of Sequence Prediction

The results of sequence prediction can be found in Table 2. For each test sequence, five different lengths (0, 1, 2, 3 and 5) of history were used. The BLEU score which is calculated between the ground truth and prediction is used as the evaluation metric.

Corpus	**Model**	**Perplexity**	**BLEU Score**				
			0	**1**	**2**	**3**	**5**
PTB	LSTM	122	0.076	0.083	0.092	0.097	0.106
	FV-LSTM	120	0.081	0.094	0.099	0.104	0.112
	MT-FV-LSTM	120	0.076	0.084	0.091	0.098	0.105
SMS	LSTM	105	0.179	0.222	0.241	0.262	0.277
	FV-LSTM	102	0.212	0.243	0.261	0.273	0.285
	MT-FV-LSTM	104	0.187	0.225	0.243	0.265	0.284

Table 2. PPL and BLEU comparison of sequence prediction task. FV-LSTM indicates the future vector enhanced LSTM, and MT-FV-LSTM indicates the future vector enhanced LSTM with multi-task training. The number below BLEU score is the length of history.

It can be observed that the PPL keeps almost the same in all the three systems. It is not surprising due to the enhanced LSTM language model is focused on the improvement of sequence level performance but PPL is a word level metric. However, the enhanced LSTM language model performs consistent better on BLEU score with different history lengths. These demonstrate that the enhanced LSTM language model can retrieve more sequence level information and get better result on sequence level metric.

To give a better understanding on the results comparison, an example has been given with the history "Japan however has", and the results of three models (traditional LSTM, enhanced LSTM, multi-task enhanced LSTM) are shown as below:

- Japan however has a N of its million;
- Japan however has been a major brand for the market;
- Japan however has been a major part of the company.

It can be observed that the enhanced LSTM language model gives more natural results on sequence prediction.

4.3. Experimental Results of N-best Rescoring

The Chinese SMS corpus is used to do speech recognition n-best rescoring. In the speech decoding stage for each audio, the sequences with the 100 highest probability will be generated. In the language model rescoring the language model score will be re-calculated by LSTM and enhanced LSTM language models, and the best path is obtained by combining both the language model score and acoustic model score. The WER comparison of n-best rescoring with different LSTM language models is given in Table 3.

It can be observed that all LSTM language models can get a large improvement over the 3-gram language model, and the new proposed LSTM language model enhanced with future vector only get a slight gain compared to the traditional LSTM language model in the single model rescoring. However, when implementing the multiple LSTM language models rescoring shown as the bottom part of Table 3, the new proposed future vector enhanced LSTM language models seem to own the huge complementary with the traditional LSTM language model. Rescoring using both the new and conventional LSTM language model together can achieve another significant improvement compared to the single LSTM language model rescoring.

Model	**WER**
3-gram	12.85
LSTM	11.39
FV	11.35
FV-MT	11.29
LSTM + FV	10.84
LSTM + FV-MT	10.75
LSTM + FV + FV-MT	10.65

Table 3. WER (%) comparison of speech recognition n-best rescoring on Chinese SMS corpus. FV indicates the future vector enhanced LSTM, and MT-FV indicates the future vector enhanced LSTM with multi-task training. All the models use equally interpolated weights.

5. CONCLUSION

Traditional LSTM language model only predicts a single word with the given history. However, LVCSR need sequence level predictions. This mismatch may cause the degradation on the performance. In this paper, a novel enhanced LSTM language model has been proposed. Enhanced LSTM language model retrieves sequence level information from future vector which is a special kind of sequence vector. Therefore enhanced LSTM language model is able to predict long term future rather than immediate word. The experiments demonstrated that the proposed enhanced LSTM language model with future vector performs well on n-best rescoring than the traditional LSTM language model, and there is a huge complementary within the new and normal LSTM language models. The results of sequence prediction also indicate that the enhanced LSTM language model can be used on other sequence level tasks.

6. DISCUSSION

Enhanced LSTM language model is an enhanced version of traditional LSTM language model, it is still a word level supervised neural network model. This is an advantage that in the pipeline of other applications, traditional LSTM language model can be straightforward replaced by enhanced LSTM language model. However, this makes the performance of enhanced LSTM language model relies on the information contains in the future vector and prediction accuracy of future vector prediction network. If the extracted future vector or predicted future vector are not generated properly, the enhanced LSTM language model system may give worse results than normal LSTM language model. Thus, the future work is listed here,

1. add gate to the network to control the scale of word level and sequence level information;
2. try other ways to extract future vector;
3. implement different methods to predict future vector;
4. use reinforcement learning to train the network directly with the sequence level evaluation metric;
5. use other sequence level tasks to test enhanced LSTM language model.

7. ACKNOWLEDGEMENT

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.

8. REFERENCES

[1] Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig, "Syntactic clustering of the web," *Computer Networks*, vol. 29, no. 8-13, pp. 1157–1166, 1997.

[2] Ted Dunning, *Statistical identification of language*, Computing Research Laboratory, New Mexico State University, 1994.

[3] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[4] Tomas Mikolov, Martin Karafiát, Lukás Burget, Jan Cernocký, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Proceedings of the 11th Annual Conference of the International Speech Communication Association, INTERSPEECH 2010*, 2010, pp. 1045–1048.

[5] X. Chen, T. Tan, Xunying Liu, Pierre Lanchantin, M. Wan, Mark J. F. Gales, and Philip C. Woodland, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association, INTERSPEECH 2015*, 2015, pp. 3511–3515.

[6] Yuening Hu, Michael Auli, Qin Gao, and Jianfeng Gao, "Minimum translation modeling with recurrent neural networks," in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014*, 2014, pp. 20–29.

[7] Thorsten Brants, Ashok C. Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean, "Large language models in machine translation," in *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, EMNLP-CoNLL 2007*, 2007, pp. 858–867.

[8] Xie Chen, Xunying Liu, Mark J. F. Gales, and Philip C. Woodland, "Recurrent neural network language model training with noise contrastive estimation for speech recognition," in *Proceedings of the 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*, 2015, pp. 5411–5415.

[9] Qi Liu, Lijuan Wang, and Qiang Huo, "A study on effects of implicit and explicit language model information for DBLSTM-CTC based handwriting recognition," in *Proceedings of the 13th International Conference on Document Analysis and Recognition, ICDAR 2015*, 2015, pp. 461–465.

[10] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba, "Sequence level training with recurrent neural networks," in *Proceedings of the 4th International Conference on Learning Representations, ICLR 2016*, 2016.

[11] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, ACL 2002*, 2002, pp. 311–318.

[12] Dietrich Klakow and Jochen Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1-2, pp. 19–28, 2002.

[13] Tianxing He, Yu Zhang, Jasha Droppo, and Kai Yu, "On training bi-directional neural network language model with noise contrastive estimation," in *Proceedings of the 10th International Symposium on Chinese Spoken Language Processing, ISCSLP 2016*, 2016.

[14] Yangyang Shi, Martha Larson, Pascal Wiggers, and Catholijn Jonker, "Exploiting the succeeding words in recurrent neural network language models," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association, INTERSPEECH 2013*, 2013, pp. 632–636.

[15] Ebru Arisoy, Abhinav Sethy, Bhuvana Ramabhadran, and Stanley Chen, "Bidirectional recurrent neural network language models for automatic speech recognition," in *Proceedings of 2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015*, 2015, pp. 5421–5425.

[16] Sam Wiseman and Alexander M. Rush, "Sequence-to-sequence learning as beam-search optimization," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016*, 2016, pp. 1296–1306.

[17] Karl Pichotta and Raymond J. Mooney, "Using sentence-level LSTM language models for script inference," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016*, 2016, pp. 279–289.

[18] Jonas Gehring, Yajie Miao, Florian Metze, and Alex Waibel, "Extracting deep bottleneck features using stacked auto-encoders," in *Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2013*, 2013, pp. 3377–3381.

[19] Jeffrey L Elman, "Finding structure in time," *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[20] Sepp Hochreiter, Yoshua Bengio, Paolo Frasconi, and Jürgen Schmidhuber, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[21] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio, "On the difficulty of training recurrent neural networks," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, 2013, pp. 1310–1318.

[22] Felix A. Gers, Nicol N. Schraudolph, and Jürgen Schmidhuber, "Learning precise timing with LSTM recurrent networks," *Journal of Machine Learning Research*, vol. 3, pp. 115–143, 2002.

[23] Alex Graves, Santiago Fernández, Faustino J. Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning, ICML 2006*, 2006, pp. 369–376.

[24] William Chan, Navdeep Jaitly, Quoc V. Le, and Oriol Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proceedings of the 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016*, 2016, pp. 4960–4964.

[25] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, 2014, pp. 1764–1772.

[26] Quoc V. Le and Tomas Mikolov, "Distributed representations of sentences and documents," in *Proceedings of the 31th International Conference on Machine Learning, ICML 2014*, 2014, pp. 1188–1196.

[27] Mohit Iyyer, Varun Manjunatha, Jordan L. Boyd-Graber, and Hal Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015*, 2015, pp. 1681–1691.

[28] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom, "A convolutional neural network for modelling sentences," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014*, 2014, pp. 655–665.

[29] Hamid Palangi, Li Deng, Yelong Shen, Jianfeng Gao, Xiaodong He, Jianshu Chen, Xinying Song, and Rabab K. Ward, "Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval," *IEEE/ACM Trans. Audio, Speech & Language Processing*, vol. 24, no. 4, pp. 694–707, 2016.

[30] Maofan Yin, Sunil Sivadas, Kai Yu, and Bin Ma, "Discriminatively trained joint speaker and environment representations for adaptation of deep neural network acoustic models," in *Proceedings of 2016 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2016*, 2016, pp. 5065–5069.

[31] Ya Xue, Xuejun Liao, Lawrence Carin, and Balaji Krishnapuram, "Multi-task learning for classification with dirichlet process priors," *Journal of Machine Learning Research*, vol. 8, pp. 35–63, 2007.

[32] Roi Reichart, Katrin Tomanek, Udo Hahn, and Ari Rappoport, "Multi-task active learning for linguistic annotations," in *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics, ACL 2008*, 2008, pp. 861–869.