DEEP RECURRENT NEURAL NETWORKS WITH LAYER-WISE MULTI-HEAD ATTENTIONS FOR PUNCTUATION RESTORATION

Seokhwan Kim

Adobe Research
San Jose, CA, USA
seokim@adobe.com

ABSTRACT

Punctuation restoration is a post-processing task of automatic speech recognition to generate the punctuation marks on unpunctuated transcripts. This paper proposes a deep recurrent neural network architecture with layer-wise multi-head attentions towards better modelling of the contexts from a variety of perspectives in putting punctuations by human writers. The experimental results show that our proposed model significantly outperforms previous state-of-the-art methods in punctuation restoration performances on IWSLT dataset.

***Index Terms*—** punctuation restoration, deep neural network, neural attention mechanism

1. INTRODUCTION

Punctuation marks play an important role in written language to organize the grammatical structures and to clarify the meaning of sentences. Consequentially, the usage of punctuations greatly affects to text readabilities and understandabilities for both human and machine readers. However, many automatic speech recognition (ASR) systems provide just a sequence of raw words with no punctuation as the transcript of a given speech input. Punctuation restoration aims to generate the punctuation marks from the unpunctuated ASR outputs, which is towards a better representation of the recognition results themselves and a higher availability of the structural features for downstream language understanding tasks based on the ASR outcomes.

While various approaches using language model [1], transition-based dependency parsing [2], and machine transition [3, 4] have been studied for punctuation restoration, the most common way to tackle this problem is defining it as a sequence labelling task to predict the punctuation label y_t for the t-th timestep in a given word sequence $X = \{x_1 \cdots x_t \cdots x_T\}$, which is formulated as follows:

$$y_t = \begin{cases} c \in C & \text{if a punctuation symbol } c \text{ is located} \\ & \text{between } x_{t-1} \text{ and } x_t, \\ \text{O} & \text{otherwise,} \end{cases}$$

where C is a closed set of punctuation symbols including 'comma', 'period' and 'question mark'.

t	x_t	y_t	t	x_t	y_t
1	so	O	12	like	O
2	if	,COMMA	13	well	?QMARK
3	we	O	14	i	,COMMA
4	make	O	15	think	O
5	no	O	16	you	O
6	changes	O	17	probably	O
7	today	O	18	already	O
8	what	,COMMA	19	have	O
9	does	O	20	the	O
10	tomorrow	O	21	picture	O
11	look	O	22	</s>	.PERIOD

Fig. 1: Example of punctuation restoration

Fig. 1 shows an example word sequence annotated with the ground-truth punctuation labels which come from a wide variety of common senses by human writers. For example, the decisive signal for inserting a comma varies according to its contexts. While the commas at $t = 2$ and 14 can be determined mostly by just the previous word at $t = 1$ and 13, the other one at $t = 8$ requires longer-term contexts to make a proper decision to place it at the boundary between two adjacent clauses. The other punctuation types, period and question mark, are nearly located at the end of each sentence. Thus, the model is supposed to find the sentence boundaries at $t = 13$ and 22 which can be detected based on short-term local patterns only. But the differentiation between period and question mark only can be done with the long-term dependencies from the beginning of each clause at $t = 8$ and 14.

Like other sequence labelling problems, conditional random fields (CRFs) had been the most successful solution for punctuation restoration in the earlier studies [5, 6, 7]. But, recently, deep neural network models have broken the records for this task's performances by using multiple fully-connected hidden layers [8], convolutional neural networks (CNNs) [8], recurrent neural networks (RNNs) [9, 10], and RNNs with neural attention mechanisms [11].

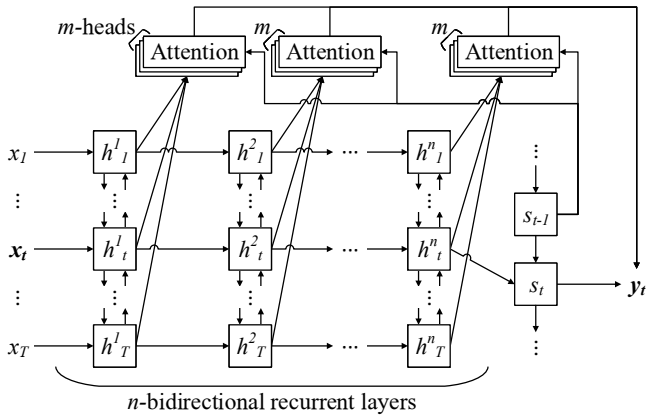


Fig. 2: Stacked RNNs with layer-wise multi-head attentions for punctuation restoration

In this work, we propose a new neural network architecture for punctuation restoration. Our model basically learns sequential contexts using RNNs followed by an attention mechanism to focus on the relevant contexts at each time step. This work has the following three main contributions from the previous studies towards improving the model's capabilities in taking various aspects into account to predict punctuations. Firstly, the model encodes the contexts not just by a single hidden recurrent layer as in [9, 11, 10], but by the stacked architecture with multiple recurrent layers to learn more hierarchical aspects. Then, neural attentions are applied not only on the top layer, but also on every intermediate hidden layer to capture the layer-wise features directly from each level of the learned hierarchy. Above all, the attentions for each layer are also diversified by multi-head attentions [12] instead of a single attention function used in [11].

2. METHOD

Our model for punctuation restoration is based on the stacked RNNs with layer-wise multi-head attentions (Fig. 2). Firstly, each word x_t in a given input sequence X is represented as a d-dimensional distributed vector $\mathbf{x}_t \in \mathbb{R}^d$. This word embedding layer can be learned from scratch with random initialization or fine-tuned from pre-trained word vectors [13, 14] during training the entire network.

Then, the sequence of the embedded word vectors is fed into bi-directional RNNs. Unlike the previous work just with a single recurrent layer [9, 11, 10], our model has a deeper architecture by stacking multiple recurrent layers on top of each other to make each layer to learn various contexts from a different perspective. Gated recurrent units (GRUs) [15] are used to get each hidden state, as follows:

$$\overrightarrow{h}^i_t = \begin{cases} \text{GRU}\left(\mathbf{x}_t, \overrightarrow{h}^1_{t-1}\right) & \text{if } i = 1, \\ \text{GRU}\left(h^{i-1}_t, \overrightarrow{h}^i_{t-1}\right) & \text{if } i = 2, \cdots n, \end{cases}$$

where $\overrightarrow{h}^i_t \in \mathbb{R}^d$ is the forward state from the beginning of the sequence to the t-th time step on the i-th recurrent layer and n is the total number of bi-directional recurrent layers. The backward state $\overleftarrow{h}^i_t$ is computed also with the same way but in the reverse order from the end of the sequence T to t. Both directional states are concatenated into the output state $h^i_t = \left[\overrightarrow{h}^i_t, \overleftarrow{h}^i_t\right] \in \mathbb{R}^{2d}$ to represent both the preceding and following contexts together.

The top-layer outputs $[h^n_1, \cdots, h^n_T]$ are forwarded to a uni-directional recurrent layer with neural attention mechanisms, as in [11]. The hidden state s_t also comes from GRU, as follows:

$$s_t = \text{GRU}\left(h^n_t, s_{t-1}\right) \in \mathbb{R}^d,$$

which represents the temporal state at each time step and constitutes a query to neural attentions. The attention mechanism used in this model is based on scaled dot-product attention [12]

$$\text{Attn}\left(Q, K, V\right) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V,$$

which is known to be more time- and memory-efficient than additive attention [16] used in [11]. More specifically, multi-head attentions [12] are applied for every layer separately to compute the weighted contexts from different representation subspaces, defined as follows:

$$f^{i,j} = \text{Attn}\left(\left(S \cdot W^{i,j}_Q\right), \left(H^i \cdot W^{i,j}_K\right), \left(H^i \cdot W^{i,j}_V\right)\right),$$

where $S = [s_1; s_2; \cdots; s_T] \in \mathbb{R}^{T \times d}$, $H^i = \left[h^i_1; h^i_2; \cdots; h^i_T\right] \in \mathbb{R}^{T \times 2d}$, $W^{i,j}_Q \in \mathbb{R}^{d \times d}$, $W^{i,j}_K \in \mathbb{R}^{2d \times d}$, and $W^{i,j}_V \in \mathbb{R}^{2d \times d}$.

Finally, the layer-wise multi-head attention outputs $f^{i,j}_t$ for all $i \in \{1, \cdots, n\}$ and $j \in \{1, \cdots, m\}$ at the t-th time step are concatenated along with s_t and fed to the fully-connected layer with softmax which generates the probabilistic distribution over the punctuation labels

$$y_t = \text{softmax}\left(\left[s_t, f^{1,1}_t, \cdots, f^{n,m}_t\right] W_y + b_y\right),$$

where $W_y \in \mathbb{R}^{nm(d+1) \times |C|}$ and $b_y \in \mathbb{R}^{|C|}$.

3. EXPERIMENTS

3.1. Data

To demonstrate the effectiveness of our proposed model, we performed experiments on the IWSLT dataset [17] which consists of the English reference transcripts of TED Talks [1]. We used the same partition of the datasets as in [8] including about 2.1M, 296k, and 13k words for training, development, and test on reference, respectively.

[1] publicly available online at https://www.ted.com/talks

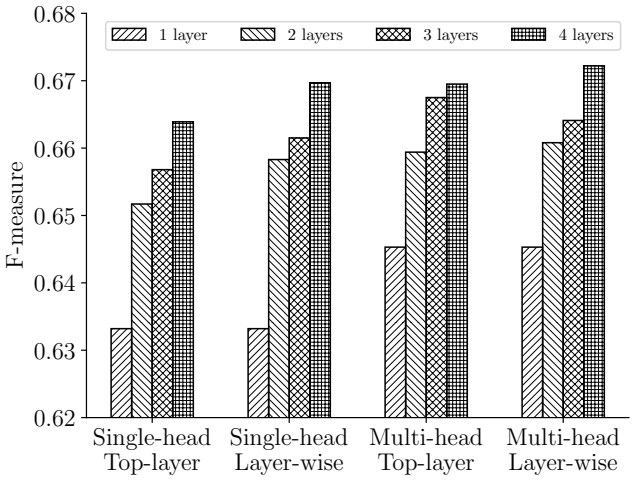


Fig. 3: Comparisons of the punctuation restoration performances of the stacked RNNs with layer-wise multi-head attentions with different network configurations

3.2. Models

Based on the dataset, we compared several models built with various combinations of different parameters and network configurations, as follows:

- Number of recurrent layers: $n \in \{1, 2, 3, 4\}$
- Number of attention heads: $m \in \{1, 2, 3, 4, 5\}$
- Attended contexts: layer-wise for all the layers or only on the top-layer

We also evaluated the impact of pre-trained word vectors in initializing the word embedding layer in comparison with random initialization. We used the 50-dimensional Glove [14] vectors trained on Wikipedia and Gigaword datasets [2].

All the models were trained with Adam optimizer [18] by minimizing the negative log likelihood loss. We set the same hidden dimension $d = 256$ across all the components except the word embedding layer initialized by pre-trained vectors. In the training phase, we used mini-batch size of 128 and applied dropout on every layer with the rate of 0.5 for regularization. We stopped the training after 100 epochs. Then, the best model was chosen based on the performance on the development set evaluated in F-measure after each epoch.

3.3. Results

The evaluations were performed on precision, recall and F-measure of the predicted labels to the ground-truth punctuations on the test dataset. In addition, we also report the overall slot error rate (SER) [19] of our proposed models compared to the previous work on the same dataset.

[2]https://nlp.stanford.edu/projects/glove/

Fig. 3 compares the performances of the models under different settings. The simplest baseline with $n = 1$ and $m = 1$ is almost equivalent to T-BRNN presented in [11], since both models have a single bi-directional recurrent layer with a single attention head. The only difference is on the implementation details of attentions between additive method [16] in T-BRNN and scaled dot-product attention [12] in ours. Our baseline model achieved 0.633 in F-measure that is slightly higher but very close to 0.631 reported as the performance of T-BRNN in the original paper [11], which was to be expected due to their structural similarities to each other.

On the other hand, all the newly proposed ideas in our model contributed to improve the punctuation restoration capabilities significantly from the baseline. Firstly, the more the number of bi-directional recurrent layers, the better performance the models achieved under every combination of the other experimental settings. For these deeper architectures, given the same number of layers, the layer-wise attentions were superior to the conventional method that attends only to the top-layers. In addition, further gains came from the diversified perspectives by our multi-head attention mechanism. As a result, the layer-wise multi-head attentive model with $n = 4$ and $m = 3$ achieved 0.672 in F-measure, which was the best performance against all the other configurations and 3.9% and 4.1% higher in absolute difference from our baseline and T-BRNN, respectively.

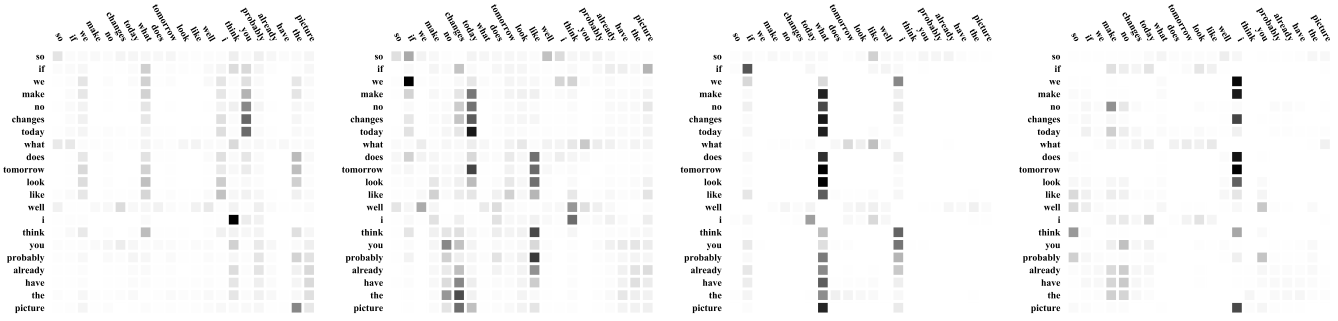
As has been shown by the improvement from T-BRNN to T-BRNN-pre in [11], our model performance was also boosted by 1.4% in absolute score when its word embedding layer was initialized not by randomization, but by the pre-trained Glove vectors. Thus, the final best performance achieved by our proposed model architecture in this experiment was 0.686 in F-measure.

Table 1 shows the detailed results of our best models in terms of per-punctuation and overall scores and also compares them to the other neural network model performances reported in recent studies on the same dataset. Overall, our proposed models outperformed all the others for almost every punctuation type and every metric. This was mainly due to the largely increased recall which was expected as a major benefit of leveraging various different aspects in capturing contexts by our proposed model components.

Previously, the best performance of punctuation restoration on the same dataset was achieved by Corr-BiRNN [10] which was based on bi-directional RNNs jointly trained not only on the punctuation, but also on the capitalization of each word in a given sequence. However, our model outperformed this previous state-of-the-art method by 3.7% in F-measure and 4.7% in SER even with no consideration of capitalization. Comparing to the best model only with the same punctuation objective, our model DRNN-LWMA-pre achieved 4.2% and 3.6% better in F-measure and SER, respectively, than T-BRNN-pre [11] where the word embedding layers were initialized by pre-trained word vectors for both models.

Table 1: Comparisons of the punctuation restoration performances with different models on test set with reference transcripts. The higher precision (P), recall (R), and F-measure (F) scores and the lower slot error rate (SER), the better results in the task. DRNN-LWMA and DRNN-LWMA-pre are our proposed models. The best score for each metric is highlighted in bold face.

	COMMA			PERIOD			QUESTION			OVERALL			
Models	P	R	F	P	R	F	P	R	F	P	R	F	SER
DNN-A [8]	48.6	42.4	45.3	59.7	68.3	63.7	-	-	-	54.8	53.6	54.2	66.9
CNN-2A [8]	48.1	44.5	46.2	57.6	69.0	62.8	-	-	-	53.4	55.0	54.2	68.0
T-LSTM [9]	49.6	41.4	45.1	60.2	53.4	56.6	57.1	43.5	49.4	55.0	47.2	50.8	74.0
T-BRNN [11]	64.4	45.2	53.1	72.3	71.5	71.9	67.5	58.7	62.8	68.9	58.1	63.1	51.3
T-BRNN-pre [11]	**65.5**	47.1	54.8	73.3	72.5	72.9	70.7	63.0	66.7	70.0	59.7	64.4	49.7
Single-BiRNN [10]	62.2	47.7	54.0	74.6	72.1	73.4	67.5	52.9	59.3	69.2	59.8	64.2	51.1
Corr-BiRNN [10]	60.9	52.4	56.4	75.3	70.8	73.0	70.7	56.9	63.0	68.6	61.6	64.9	50.8
DRNN-LWMA	63.4	55.7	59.3	76.0	73.5	74.7	**75.0**	**71.7**	**73.3**	**70.0**	64.6	67.2	47.3
DRNN-LWMA-pre	62.9	**60.8**	**61.9**	**77.3**	**73.7**	**75.5**	69.6	69.6	69.6	69.9	**67.2**	**68.6**	**46.1**



(a) First layer (b) Second layer (c) Third layer (d) Fourth layer

Fig. 4: Visualization of sample attentions on the word sequence in Fig. 1. Each plot shows the pair-wise weights from the first attention head of each layer in our best model. The x-axis and y-axis correspond to the words to be attended and to be predicted, respectively. The higher the weight of a word pair, the darker the cell at their intersection.

Fig. 4 visualizes the attention weights by multiple heads from our best model in restoring the punctuations on the input sequence in Fig. 1. This generally shows that all the heads mostly attended to the boundaries of syntactic units including sentences and clauses, which is meant to be a typical behavior of human writers in putting punctuations. Furthermore, each head produced a unique distribution of attending to the contexts which is clearly differentiated from the others. More specifically, the weights from the first layer attentions (Fig. 4a) spread over the syntactic boundaries in the whole sequence. In contrast, each of the others from higher layers tends to have a specific focus on particular spots in the sequence. For example, Fig. 4b indicates high attentions at the end of the clauses and sentences, while Fig. 4c and Fig. 4d concentrate on the beginning of each syntactic unit. These observations demonstrate the capabilities of our model in capturing various aspects separately from each other by the distributed attentions across multiple layers and multiple heads in the proposed architecture.

4. CONCLUSIONS

This paper presented a deep neural network architecture for punctuation restoration. The model is based on stacked RNNs with layer-wise multi-head attentions, which aims to better incorporate various aspects into context learning. Experimental results showed that the proposed model contributed to improve the punctuation restoration performance on IWSLT dataset with respect to previous state-of-the-art models.

Furthering this work, we have been considering the following two directions as our next steps: improving our model by learning both from lexical and prosodic features; and exploring the effectiveness of our model for the joint prediction of punctuation and capitalization.

5. ACKNOWLEDGEMENTS

We would like to thank Ottokar Tilk for providing the scripts for data preprocessing and model evaluation used in [11].

6. REFERENCES

[1] Agustin Gravano, Martin Jansche, and Michiel Bacchiani, "Restoring punctuation and capitalization in transcribed speech," in *Proceedings of the 34th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2009, pp. 4741–4744.

[2] Dongdong Zhang, Shuangzhi Wu, Nan Yang, and Mu Li, "Punctuation prediction with transition-based parsing," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, 2013, vol. 1, pp. 752–760.

[3] Stephan Peitz, Markus Freitag, Arne Mauser, and Hermann Ney, "Modeling punctuation prediction as machine translation," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2011, pp. 238–245.

[4] Eunah Cho, Jan Niehues, Kevin Kilgour, and Alex Waibel, "Punctuation insertion for real-time spoken language translation," in *Proceedings of the 12th International Workshop on Spoken Language Translation (IWSLT)*, 2015, pp. 173–179.

[5] Wei Lu and Hwee Tou Ng, "Better punctuation prediction with dynamic conditional random fields," in *Proceedings of the 2010 conference on empirical methods in natural language processing (EMNLP)*, 2010, pp. 177–186.

[6] Xuancong Wang, Hwee Tou Ng, and Khe Chai Sim, "Dynamic conditional random fields for joint sentence boundary and punctuation prediction," in *Proceedings of the 13th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2012, pp. 1384–1387.

[7] Nicola Ueffing, Maximilian Bisani, and Paul Vozila, "Improved models for automatic punctuation prediction for spoken and written text.," in *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech)*, 2013, pp. 3097–3101.

[8] Xiaoyin Che, Cheng Wang, Haojin Yang, and Christoph Meinel, "Punctuation prediction for unsegmented transcript based on word vector.," in *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC)*, 2016, pp. 654–658.

[9] Ottokar Tilk and Tanel Alumäe, "LSTM for punctuation restoration in speech transcripts," in *Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2015, pp. 683–687.

[10] Vardaan Pahuja, Anirban Laha, Shachar Mirkin, Vikas Raykar, Lili Kotlerman, and Guy Lev, "Joint learning of correlated sequence labelling tasks using bidirectional recurrent neural networks," in *Proceedings of the 18th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2017, pp. 548–552.

[11] Ottokar Tilk and Tanel Alumäe, "Bidirectional recurrent neural network with attention mechanism for punctuation restoration.," in *Proceedings of the 17th Annual Conference of the International Speech Communication Association (INTERSPEECH)*, 2016, pp. 3047–3051.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems (NIPS)*, 2017, pp. 5998–6008.

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

[14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[15] Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, "Learning phrase representations using rnn encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1724–1734.

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

[17] Marcello Federico, Mauro Cettolo, Luisa Bentivogli, Paul Michael, and Stüker Sebastian, "Overview of the IWSLT 2012 evaluation campaign," in *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*, 2012, pp. 12–33.

[18] Diederik P Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[19] John Makhoul, Francis Kubala, Richard Schwartz, Ralph Weischedel, et al., "Performance measures for information extraction," in *Proceedings of DARPA broadcast news workshop*, 1999, pp. 249–252.